

Cursor Desktop Classic IDE: Nihai Teknik Referans Belgesi

Sürüm: 2026.08 — Nihai (Ultimate)

Dil: Türkçe

Kapsam: Cursor Desktop **Classic IDE (Editor Window)** — tam teknik mimari referansı

Not: Agents Window ile **aynı backend, aiserver.v1 API ve AI pipeline** paylaşılır.

Doğrulama: Resmi kaynak docs.cursor.com ve cursor.com/blog. [TERS MÜHENDİSLİK] / [GÖZLEM] etiketli maddeler sürüm bağımlıdır.

Alan	Değer
Kelime hedefi	15.000+
Ana bölüm	27 + 5 ek
Mermaid	12+
Tablo	35+

İçindekiler

- [Giriş, Kapsam ve Metodoloji](#)
- [Classic IDE ve Agents Window](#)
- [Katmanlı Sistem Mimarisi](#)
- [Electron Süreç Modeli ve IPC](#)
- [Connect-RPC ve aiserver.v1 Protokol Katmanı](#)
- [Tab: Pipeline, Model ve RL Matematiği](#)
- [Fast Apply ve Speculative Edits](#)
- [Inline Edit \(Cmd+K\)](#)
- [Agent Orkestrasyonu ve Model Harness](#)
- [Agent Araçları: Instant Grep vs ripgrep](#)
- [Shadow Workspace ve gRPC IPC](#)
- [Kod Tabanı İndeksleme: Merkle, Simhash, Content Proof](#)
- [Turbopuffer Vektör Deposu](#)
- [Dynamic Context Discovery](#)
- [Context Ring, Compaction ve preCompact Hook](#)
- [Rules, Skills, Hooks ve MCP](#)
- [Agent Sandbox: Seatbelt ve Landlock](#)
- [BYOK, Kimlik Doğrulama ve state.vscdb](#)
- [Composer RL ve 5 Saatlik Üretim Döngüsü](#)
- [Bağlam Derleme ve Model Yönlendirme](#)
- [Gizlilik, Güvenlik ve Veri Yönetimi](#)
- [Git Entegrasyonu ve Attribution](#)
- [VS Code Uyumluluğu ve Dağıtım](#)
- [Performans Profili ve Kapasite Planlama](#)
- [Operasyonel En İyi Uygulamalar](#)
- [Sık Sorulan Sorular](#)
- [Sonuç ve Mimari Çıkarımlar](#)

28. [Derinlemesine Teknik Ekler](#)

- [Ek A: Terimler Sözlüğü](#)
- [Ek B: Sürüm Matrisi](#)
- [Ek C: Veri Akış Diyagramları](#)
- [Ek D: Protobuf ve SQL](#)
- [Ek E: Doğrulama Matrisi](#)

1. Giriş, Kapsam ve Metodoloji

Cursor Desktop, Visual Studio Code (VS Code) fork'u üzerine inşa edilmiş, yapay zeka odaklı bir geliştirme platformudur. **Classic IDE** (resmi adıyla **Editor Window**), geleneksel editör merkezli düzeni korur: Explorer, sekmeli editör, terminal ve isteğe bağlı Agent paneli. Cursor 3.x ile birlikte varsayılan açılış **Agents Window** olabilir; ancak günlük kod yazımı, debugging ve dosya odaklı iş akışları için Classic IDE hâlâ birincil yüzeydir.

Bu belge, Classic IDE modunun **nihai teknik referansıdır**. Önceki `cursor-desktop-classic-ide-tr.md` belgesinin tamamını kapsar, genişletir ve aşağıdaki ultra-teknik konuları birleştirir:

Konu	Kaynak türü
Electron IPC kanalları	Resmi VS Code mimarisi + [GÖZLEM]
Connect-RPC / aiserver.v1	[TERS MÜHENDİSLİK] (paket çıkarımı)
Tab RL ödül fonksiyonu	Resmi Cursor blog
Fast Apply / speculative edits	Resmi Cursor blog
Merkle + simhash + content proof	Resmi Cursor blog
Turbopuffer namespace modeli	Resmi blog + [GÖZLEM]
Agent harness (model başına)	Resmi Cursor blog
Dynamic context paths	Resmi Cursor blog
Shadow Workspace gRPC	Resmi Cursor blog
Seatbelt / Landlock sandbox	Resmi Cursor blog
state.vscdb anahtarları	[TERS MÜHENDİSLİK] + resmi hooks docs
Instant Grep vs ripgrep	Resmi Cursor blog (Mart 2026)
preCompact hook	Resmi Cursor docs
Context ring / compaction	Resmi blog + [GÖZLEM]
Composer RL 5 saat döngüsü	Resmi Cursor blog

1.1. Metodoloji ve doğrulama etiketleri

Belgede üç güven seviyesi kullanılır:

1. **Resmi**: [cursor.com/blog](#), [docs.cursor.com](#), MCP spesifikasyonu.
2. **[GÖZLEM]** : Davranış gözlemi; sürüm bağımlı.

3. **[TERS MÜHENDİSLİK]** : Minified bundle, unite.proto, topluluk istemcileri (eisbaw/cursor_api_demo, wisdog/cursor-tab).

Kurumsal değerlendirme için: güvenlik ve veri akışı kararlarında yalnızca **Resmi** etiketli maddelere güvenin; diğerleri mühendislik keşfi içindir.

1.2. Paylaşılan backend: Classic IDE vs Agents Window

Classic IDE ve Agents Window **aynı Electron binary, aynı Extension Host, aynı aiserver.v1 backend** (`api2.cursor.sh`) ve **aynı indeksleme servisini** kullanır. Fark yalnızca renderer layout ve varsayılan odak noktasıdır:

```
graph LR
    subgraph "Tek Cursor Binary"
        CW["Classic IDE<br/>Editor Window"]
        AW["Agents Window"]
    end
    subgraph "Paylaşılan Katman"
        EH["Extension Host"]
        AI["AI Orkestrasyon"]
        IDX["İndeksleme"]
    end
    subgraph "Bulut"
        API["api2.cursor.sh<br/>aiserver.v1"]
    end
    CW --> EH
    AW --> EH
    EH --> AI
    AI --> IDX
    AI --> API
    IDX --> API
```

Agents Window'da bulunan Background Agent başlatma, uzaktan oturum yönetimi gibi özellikler Classic IDE'den de erişilebilir; yalnızca UI yerleşimi değişir.

2. Classic IDE ve Agents Window

2.1. Pencere türleri karşılaştırması

Pencere	Odak	Klavye (macOS / Win)	Terminal bayrağı
Editor Window (Classic IDE)	Kod editörü	Shift+Cmd+N / Shift+Ctrl+N	<code>cursor . --classic</code>
Agents Window	Sohbet / görev	Varsayılan (3.x+)	<code>cursor .</code>

Classic IDE düzeni: Activity Bar → Explorer → Editor Area → Panel (Terminal) → Secondary Side Bar (Agent).

2.2. Geçiş yöntemleri

- Shift+Cmd+N / Shift+Ctrl+N — Open Editor Window
- Command Palette → "Open Editor Window"
- `cursor . --classic`
- Agents Window üst çubuk IDE düğmesi

5. Window Restoration: son oturumda Editor Window açıkken `Cmd+Q` ile tam çıkış

2.3. Oturum geri yükleme (3.11+)

"Open Agents Window on Startup" kapalıyken Cursor **son pencere oturumunu** geri yükler. Kalıcı Classic IDE için yalnızca Editor Window açıkken çıkış yapın.

2.4. Teknik konum

Classic IDE, sunum katmanının tam kapsamlı varyantıdır. Tab pipeline, Fast Apply, Agent harness, MCP client, indeksleme ve sandbox — hepsi pencere türünden bağımsızdır.

3. Katmanlı Sistem Mimarisi

Cursor Desktop Classic IDE monolitik bir LLM sarmalayıcısı değil; **çok katmanlı orkestrasyon platformudur**.

```
graph TB
    subgraph "Sunum (Classic IDE Renderer)"
        UI["Explorer / Editor / Terminal"]
        TAB_UI["Tab Ghost Text"]
        AGENT_UI["Agent Panel Cmd+I/L"]
    end
    subgraph "Orkestrasyon (Cursor Extension Layer)"
        CTX["Context Assembly"]
        HARNESS["Agent Harness<br/>model başına prompt+tools"]
        RULES["Rules Engine"]
        TOOLS["Tool Orchestrator"]
        IDX["Index + Instant Grep"]
        MCP_C["MCP Client"]
        SW["Shadow Workspace Client"]
    end
    subgraph "Süreçler"
        MAIN["Main Process"]
        EH["Extension Host"]
        SHADOW_EH["Shadow Extension Host"]
    end
    subgraph "Bulut api2.cursor.sh"
        RPC["Connect-RPC aiserver.v1"]
        EMB["Embedding Service"]
        TP["Turbopuffer"]
    end
    UI --> CTX
    TAB_UI --> RPC
    AGENT_UI --> HARNESS
    HARNESS --> TOOLS
    CTX --> RULES
    TOOLS --> MCP_C
    TOOLS --> SW
    SW --> SHADOW_EH
    EH --> IDX
    IDX --> EMB --> TP
    HARNESS --> RPC
```

3.1. Mimari ilkeler

İlke	Açıklama
İstemci tarafı araç yürütme	LLM önerir; Cursor yürütür ve onaylar
Durumsuz model, durumlu oturum	Her istekte tam bağlam gönderilir
Yetenek spektrumu	Tab (~320ms) → Cmd+K → Agent (saniyeler-dakikalar)
VS Code fork + AI katmanı	LSP/DAP çıktıları AI bağlamına beslenir
Dynamic context	Statik enjeksiyon yerine dosya tabanlı keşif

3.2. Veri sınıflandırması

Sınıf	Örnekler	Konum
Yerel plaintext	Açık dosyalar, terminal çıktısı	Disk / bellek
Şifreli bulut metadata	Embedding, obfuscated path	Turbopuffer
Kimlik bilgisi	JWT, BYOK API key	state.vscdb (safeStorage)
Geçici inference	Prompt + yanıt	TLS ile api2.cursor.sh

4. Electron Süreç Modeli ve IPC

Classic IDE, Electron çok süreçli mimarisini VS Code'dan devralır.

4.1. Süreç haritası

Süreç	Rol	Classic IDE bileşenleri
Main	Pencere yaşam döngüsü, native OS, güncelleme	Editor Window oluşturma
Renderer	UI render	Explorer, Editor, Tab overlay, Agent chat UI
Extension Host	Node.js izolasyonu	LSP, linter, Cursor AI extension
Shared Process	Arka plan görevleri	File watcher, yerel arama indeksi
GPU / Utility	Chromium yardımcıları	Render hızlandırma

4.2. VS Code IPC kanalları [GÖZLEM / VS Code kaynak]

Renderer ↔ Extension Host iletişimi MessagePort tabanlıdır. Cursor, bu kanalı Shadow Workspace için yeniden kullanır.

Kanal / protokol	Yön	İçerik
------------------	-----	--------

vscode:message (MessagePort)	Renderer → Extension Host	Komutlar, editör olayları
Extension Host RPC	Bidirectional	\$executeCommand, \$provideDiagnostics
cursor.ai.* [GÖZLEM]	Renderer → Cursor ext	Tab/Agent istekleri (internal API)
File watcher events	Shared → EH	İndeksleme tetikleyicisi

Not: Tam kanal adları minified bundle'da obfuscate edilmiştir; tablo [GÖZLEM] ve VS Code kaynak kodu eşlemesine dayanır.

4.3. Shadow Workspace IPC genişlemesi

Electron sandboxing nedeniyle iki renderer doğrudan konuşamaz. Cursor'ın çözümü (resmi blog):

```
sequenceDiagram
    participant R1 as Classic IDE Renderer
    participant EH1 as Extension Host (görünür)
    participant EH2 as Extension Host (shadow)
    participant R2 as Shadow Renderer (gizli)

    R1->>EH1: VS Code MessagePort IPC
    EH1->>EH2: Bağımsız IPC (gRPC over buf)
    EH2->>R2: Extension komutları
    R2-->>EH2: LSP diagnostics
    EH2-->>EH1: gRPC yanıt
    EH1-->>R1: Lint / doğrulama sonucu
```

Extension Host'tan Extension Host'a **gRPC + Protocol Buffers** kullanımı, VS Code'un JSON RPC serileştirmesine kıyasla daha sürdürülebilir ve tip güvenli bir yol sağlar (Bölüm 11).

4.4. Responsiveness tasarımı

Tab önerileri ve Agent tool call'ları ana UI thread'i bloke etmez. Ağır işler Extension Host veya arka plan worker'larda yürütülür; renderer yalnızca sonuçları görselleştirir.

5. Connect-RPC ve aiserver.v1 Protokol Katmanı

Classic IDE'nin tüm bulut AI iletişimi **Connect-RPC** üzerinden `api2.cursor.sh` adresine gider. Bu, gRPC-Web alternatifi Buf Connect protokolüdür.

5.1. Transport özeti [TERS MÜHENDİSLİK]

Parametre	Değer
Base URL	<code>https://api2.cursor.sh</code>
Protokol	Connect-RPC (HTTP/2)
Content-Type	<code>application/connect+proto</code>
Connect-Protocol-Version	1

Encoding	Binary envelope [type:1][len:4BE][payload]
----------	--

5.2. Zorunlu HTTP başlıkları [TERS MÜHENDİSLİK]

Header	Açıklama
Authorization	Bearer {JWT} — state.vscdb'den
x-cursor-client-version	İstemci sürümü (ör. 3.0.16)
x-cursor-client-type	ide
x-cursor-client-os / -arch	Platform bilgisi
x-cursor-checksum	{cipher_timestamp}{machine_id}
x-cursor-config-version	UUID
x-cursor-timezone	IANA timezone
x-ghost-mode	Privacy Mode bayrağı
x-session-id	Token'dan türetilmiş UUID v5
x-request-id	İstek UUID

5.3. aiserver.v1 servisleri [TERS MÜHENDİSLİK – unite.proto]

```
syntax = "proto3";
package aiserver.v1;

service AiService {
  rpc StreamCpp(StreamCppRequest) returns (stream StreamCppResponse);
  rpc StreamNextCursorPrediction(StreamNextCursorPredictionRequest)
    returns (stream StreamNextCursorPredictionResponse);
  rpc CppConfig(CppConfigRequest) returns (CppConfigResponse);
  rpc RefreshTabContext(RefreshTabContextRequest) returns
    (RefreshTabContextResponse);
  rpc RecordCppFate(RecordCppFateRequest) returns (RecordCppFateResponse);
}

service CppService {
  rpc MarkCppForEval(MarkCppRequest) returns (MarkCppResponse);
  rpc RecordCppFate(RecordCppFateRequest) returns (RecordCppFateResponse);
}

service FileSyncService {
  rpc FSUploadFile(FSUploadFileRequest) returns (FSUploadFileResponse);
  rpc FSSyncFile(FSSyncFileRequest) returns (FSSyncFileResponse);
  rpc FSGetFileContents(FSGetFileContentsRequest) returns
    (FSGetFileContentsResponse);
}
```

```

service CursorPredictionService {
  rpc CursorPredictionConfig(CursorPredictionConfigRequest)
    returns (CursorPredictionConfigResponse);
}

```

Agent sohbeti için [TERS MÜHENDİSLİK] :

RPC	Servis	Akış
StreamUnifiedChatWithTools	ChatService	Server streaming
StreamChat	AIService	Server streaming
AvailableModels	AIService	Unary
GetUsableModels	AIService	Unary

5.4. Tab isteği mesaj yapısı (özet) [TERS MÜHENDİSLİK]

```

message StreamCppRequest {
  CurrentFileInfo current_file = 1;
  repeated CodeBlock additional_files = 2;
  ModelDetails model_details = 5;
  repeated LinterErrors linter_errors = 7;
  // edit history, workspace root, git context...
}

message CurrentFileInfo {
  string relative_workspace_path = 1;
  string contents = 2;
  optional string sha_256_hash = 17;
  CursorPosition cursor_position = 3;
  repeated Diagnostic diagnostics = 7;
  string workspace_root_path = 19;
}

message StreamCppResponse {
  string text = 1;          // diff / completion
  optional CursorPrediction prediction = 2;
}

```

RecordCppFate RPC'si Tab RL döngüsü için kabul/red sinyallerini sunucuya iletir (Bölüm 6).

5.5. Connect-RPC çerçeve formatı

```

graph LR
  REQ["HTTP/2 POST"] --> ENV["Connect Envelope"]
  ENV --> T["type byte"]
  ENV --> L["length 4B BE"]
  ENV --> P["protobuf payload"]

```

```
P --> SRV["aiserver.v1 handler"]
SRV --> RES["stream envelopes"]
```

Unary çağrılar tek envelope; streaming çağrılar (Tab, Agent) ardışık envelope dizisi döndürür.

6. Tab: Pipeline, Model ve RL Matematiği

Cursor Tab, 8B parametrelili özel model ile çalışan, düşük gecikmeli (320 ms hedef) otomatik tamamlama sistemidir. Token tahmininden ziyade **bir sonraki düzenleme eylemini** tahmin eder.

6.1. Pipeline

```
sequenceDiagram
    participant ED as Editör
    participant TAB as Tab Engine
    participant API as api2.cursor.sh
    participant LSP as Language Server

    ED->>TAB: keystroke + cursor + file hash
    TAB->>LSP: diagnostics
    LSP-->>TAB: linter errors
    TAB->>API: StreamCpp (Connect-RPC)
    API-->>TAB: diff suggestion stream
    TAB->>ED: ghost text
    ED->>API: RecordCppFate (accept/reject)
```

6.2. Bağlam kaynakları

Kaynak	Tab'a etki	Rules etkisi
Mevcut dosya	Evet	Hayır
Edit chain (oturum geçmişi)	Evet	Hayır
LSP diagnostics	Evet	Hayır
Cross-file portal	Evet	Hayır
Project Rules	—	Hayır

6.3. Online RL ödül fonksiyonu (Resmi — cursor.com/blog/tab-rl)

Tab, **policy gradient** ile eğitilir. Öneri gösterilip gösterilmeme kararı modele entegre edilmiştir.

Ödül tablosu:

Olay	Ödül (r)
Kullanıcı kabul eder (Tab)	(+0.75)
Kullanıcı reddeder (Escape)	(-0.25)
Öneri gösterilmez (sessizlik)	(0)

Beklenen ödül: Öneri gösterildiğinde kabul olasılığı (p) ise:

$$[E[r \mid \text{göster}]] = 0.75p - 0.25(1-p) = p - 0.25]$$

($E[r] > 0$) ancak ve ancak ($p > 0.25$). Optimal politika: tahmini kabul olasılığı %25'in üzerindeyse öner, aksi halde sessiz kal.

Sonuç metrikleri (resmi blog): %21 daha az öneri, kalan önerilerde %28 daha yüksek kabul oranı.

6.4. RL döngü zamanlaması

Aşama	Tab [GÖZLEM]	Composer (karşılaştırma)
Deploy checkpoint	Saatler içinde	~5 saat
Veri toplama	On-policy kullanıcı etkileşimi	Üretim token'ları
Retrain	Günlük birden çok checkpoint	Günde birden çok
RPC geri bildirim	RecordCppFate, MarkCppForEval	Agent reward sinyalleri

Tab RL döngüsü ~1.5–2 saat olarak raporlanmıştır (üçüncü taraf analiz); resmi blog "gün içinde birden çok kez" ifadesini kullanır.

6.5. Jump-in-file ve Cursor Prediction

`StreamNextCursorPrediction` RPC'si kabul sonrası bir sonraki imleç konumunu tahmin eder.
`CursorPredictionService.CursorPredictionConfig` eşikleri yapılandırır.

6.6. Tab kontrol

Eylem	macOS	Win/Linux
Kabul	Tab	Tab
Red	Escape	Escape
Kelime kabul	Cmd+→	Ctrl+→
Snooze / Disable	Tab göstergesi	Tab göstergesi

6.7. Ek Tab detayları (Classic IDE referans)

6.1. Tab Nedir?

Cursor Tab, yazarken gerçek zamanlı kod önerileri sunan AI destekli otomatik tamamlama sistemidir. Geleneksel IntelliSense'ten farklı olarak Tab yalnızca bir sonraki token'ı tahmin etmez; **bir sonraki düzenleme konumunu, çok satırlı değişiklikleri, eksik import ifadelerini** ve hatta **başka dosyalardaki koordineli düzenlemeleri** önerebilir.

Tab, Classic IDE'nin günlük kullanımdaki en sık temas edilen AI bileşenidir. Gecikme hedefi ~320 ms civarındadır (GitHub Copilot'un ~890 ms ortalamasına kıyasla belirgin avantaj).

6.2. Tab'ın Teknik Pipeline'ı

```
sequenceDiagram
    participant DEV as Geliştirici
    participant ED as Editör (Classic IDE)
    participant TAB as Tab Engine
    participant CTX as Bağlam Derleyici
    participant MODEL as Tab Modeli (~8B param)
    participant LSP as Language Server

    DEV->>ED: Tuş vuruşu / düzenleme
    ED->>CTX: Dosya içeriği + imleç + son düzenlemeler
    CTX->>LSP: Linter/diagnostic sorgusu
    LSP-->>CTX: Hata/uyarı listesi
    CTX->>TAB: Derlenmiş bağlam
    TAB->>MODEL: Tamamlama isteği
    MODEL-->>TAB: Öneri (çok satırlı olabilir)
    TAB->>ED: Gri metin olarak göster
    DEV->>ED: Tab (kabul) veya Escape (red)
    ED->>ED: Öneriyi uygula
    DEV->>ED: Tekrar Tab → jump-in-file
    TAB->>MODEL: Sonraki düzenleme konumu tahmini
```

6.3. Tab'ın Bağlam Kaynakları

Tab önerileri şu sinyalleri kullanır:

- **Mevcut dosyanın tamamı** (veya token bütçesi içindeki bölüm)
- **Son düzenlemeler** — oturum içinde yapılan değişiklikler (Tab "edit chain" takibi yapar)
- **Linter/language server hataları** — eksik import, tip uyumsuzluğu vb.
- **Çevredeki kod** — imleç öncesi/sonrası token'lar
- **Proje geneli sinyaller** — cross-file edit tahminleri için

Rules **Tab'ı etkilemez** — bu önemli bir ayrımdır. Tab, Rules'tan bağımsız çalışır.

6.4. Tab Özellikleri

6.4.1. Çok Satırlı Tamamlama

Tab tek satır değil, birden fazla satırı kapsayan öneriler sunabilir. Fonksiyon gövdeleri, sınıf tanımları, test iskeletleri gibi yapılar tek Tab ile kabul edilebilir.

6.4.2. Otomatik Import

TypeScript, Python ve diğer dillerde çözülmemiş semboller için Tab eksik import ifadelerini otomatik ekleyebilir. Bu, LSP tanımlayıcı bilgisi + model tahmininin birleşimidir.

6.4.3. Jump-in-File

Bir Tab önerisi kabul edildikten sonra **tekrar Tab** tuşuna basmak, Tab'ın tahmin ettiği **bir sonraki düzenleme konumuna** imleci taşır. Bu "edit chain" özelliği, geliştiricinin dosya içinde manuel navigasyon yapmadan ardışık düzenlemeler yapmasını sağlar.

6.4.4. Cross-File Edits (Dosyalar Arası Düzenleme)

Tab, bir dosyadaki deęiřiklięin bařka bir dosyada gncelleme gerektirdięini tahmin edebilir. Bu durumda editrn alt kısmında bir **portal penceresi** belirir; geliřtirici ilgili dosyaya geip nerilen deęiřiklięi kabul edebilir.

6.5. Tab Kontrol ve Yapılandırma

Eylem	Kısayol (macOS)	Kısayol (Win/Linux)
Tam neriyi kabul et	Tab	Tab
neriyi reddet	Escape	Escape
Kelime kelime kabul	Cmd+→	Ctrl+→

Tab durumu saę alt kředeki **Tab gstergesinden** ynetilir:

- **Snooze:** Geici olarak devre dıřı bırak (sre seilebilir)
- **Disable globally:** Tm dosyalarda kapat
- **Disable for extensions:** Belirli dosya trlerinde kapat (markdown, JSON vb.)

Cursor Settings → Tab altında ek yapılandırmalar mevcuttur. "Accept Cursor Tab Suggestions" kısayolu Keyboard Shortcuts'tan yeniden eřlenebilir.

6.6. Tab vs Dięer AI zellikleri

Tab, **hız-ncelikli, dřk gecikmeli, tahmin odaklı** bir sistemdir. Mimari olarak Agent'tan tamamen ayrı bir model ve pipeline kullanır. Tab ~8 milyar parametrelili zel bir modelle alıřır; Agent ise frontier modelleri (Claude, GPT, Composer vb.) kullanır.

Ne zaman Tab kullanılmalı:

- Boilerplate kod yazımı
- Tekrarlayan kalıplar
- Bilinen API'lerle alıřma
- Import/signature tamamlama
- Test iskeleti oluřturma

Ne zaman Tab yeterli deęildir:

- Mimari kararlar
- ok dosyalı refactoring
- Karmařık iř mantıęı deęiřiklikleri
- Baęlaman proje genelinde anlařılması gereken durumlar

7. Fast Apply ve Speculative Edits

Fast Apply (Instant Apply), Agent ve Inline Edit'in dosya dzenlemelerini uygularken kullandıęı zel inference pipeline'ıdır. Frontier model "taslak diff" retir; **llama-70b-ft-spec** [GZLEM] sınıfı model bunu tam dosyaya hızla uygular.

7.1. Mimari ayırım

```
graph LR
  LLM["Frontier LLM<br/>taslak deęiřiklik"] --> APPLY["Fast Apply Model<br/>~70B"]
```

```
fine-tuned"]
  ORIG["Orijinal dosya"] --> APPLY
  APPLY --> DIFF["Streaming diff<br/>~1000 tok/s"]
  DIFF --> ED["Classic IDE Editor"]
```

7.2. Speculative edits algoritması (Resmi)

Klasik speculative decoding küçük bir draft model kullanır. Kod düzenlemede **draft model gerekmez — mevcut dosya draft'tır**.

Adım	İşlem
1	Orijinal dosyayı chunk'lara böl
2	Chunk'ları "speculated output" olarak modele ver
3	Model değişmeyen chunk'ları toplu onaylar (paralel)
4	Ayrışma noktasında yeni token üret
5	Kalan değişmemiş koddan spekülasyona devam et

Performans (resmi): ~~4000 token/s~~ (3500 char/s), vanilla Llama-3-70B'ye göre ~13×, GPT-4 tabanlı yaklaşıma göre ~9×.

7.3. Fireworks AI entegrasyonu

Fast Apply, Fireworks inference engine üzerinde custom speculation API ile deploy edilir. Classic IDE istemcisi diff'i stream ederken kullanıcı incelemeye başlayabilir.

7.4. Fast Apply vs tam dosya rewrite

Yaklaşım	Gecikme	Doğruluk profili
Tam dosya rewrite (GPT-4)	Yüksek	İyi
Search-replace diff	Orta	Orta (büyük dosyada kırılğan)
Fast Apply + speculative edits	Düşük	Frontier'tan iyi [resmi benchmark]

7.5. Classic IDE'de görünüm

Inline Edit (Cmd+K) ve Agent file edit araçları diff renderer kullanır: kırmızı silinen, yeşil eklenen. Fast Apply arka planda çalışır; kullanıcı yalnızca stream edilen diff'i görür.

8. Inline Edit (Cmd+K)

Inline Edit, chat paneli açmadan editörde hedefli düzenleme yapar. Fast Apply pipeline'ını kullanabilir.

```
sequenceDiagram
    participant DEV as Geliştirici
    participant IE as Inline Edit Engine
    participant LLM as Frontier LLM
    participant FA as Fast Apply
```

```
DEV->>IE: Seçim + Cmd+K + talimat
IE->>LLM: seçili kod + bağlam
LLM-->>IE: taslak patch
IE->>FA: taslak + tam dosya
FA-->>IE: streaming diff
IE->>DEV: kabul / red / follow-up
```

Özellik	Inline Edit	Agent
Gecikme	~1-3 sn	~5-60+ sn
Kapsam	Tek dosya	Çok dosya + terminal
Rules	Hayır	Evet
Fast Apply	Evet	Evet (edit araçları)

Soru modu: Opt+Return / Alt+Return — soru sor, "do it" ile uygula.

Agent geçişi: Seçim + Cmd+L → Agent bağlamına aktar.

8. Cmd+K: Satır İçi Düzenleme (Inline Edit)

8.1. Inline Edit Mimarisi

Inline Edit (Cmd+K / Ctrl+K), chat panelini açmadan doğrudan editörde hedefli kod değişiklikleri yapmayı sağlar. Cerrahi, tek dosya kapsamlı, düşük gecikmeli bir AI düzenleme aracıdır.

```
sequenceDiagram
    participant DEV as Geliştirici
    participant ED as Editör
    participant IE as Inline Edit Engine
    participant MODEL as LLM
    participant DIFF as Diff Renderer

    DEV->>ED: Kod seç
    DEV->>ED: Cmd+K
    DEV->>IE: Talimat gir ("async function'a çevir")
    IE->>MODEL: Seçili kod + talimat + dosya bağlamı
    MODEL-->>IE: Düzenlenmiş kod
    IE->>DIFF: Diff göster (kırmızı=silinen, yeşil=eklenen)
    DEV->>DIFF: Kabul / Red / Follow-up
```

8.2. Kullanım Akışı

- Değiştirilecek kodu seçin
- Cmd+K (macOS) veya Ctrl+K (Windows/Linux) basın
- Talimatınızı yazın (ör. "Bu fonksiyonu async/await kullanacak şekilde refactor et")
- Return ile uygulayın
- Follow-up talimatları ile iteratif iyileştirme yapın

8.3. Soru Modu (Question Mode)

Inline Edit üzerinden hızlı soru sormak mümkündür:

- Kodu seçin → `Cmd+K`
- `Opt+Return` (macOS) veya `Alt+Return` (Win/Linux) ile soru moduna geçin
- Sorunuzu yazın
- Önerilen değişikliği uygulamak için "do it" yazıp Return basın

8.4. Inline Edit vs Agent Geçiş

Çok dosyalı veya karmaşık değişiklikler için:

- Kodu seçin → `Cmd+L` (macOS) / `Ctrl+L` (Win/Linux)
- Seçili kod Agent bağlamına aktarılır
- Agent panelinde çok dosyalı düzenleme başlatılır

8.5. Inline Edit'in Bağlam Profili

Inline Edit şu bağlamı kullanır:

- Seçili kod bloğu
- Mevcut dosyanın tamamı (token bütçesi dahilinde)
- LSP tanımlayıcıları ve linter çıktıları
- Rules uygulanmaz** (User Rules dahil — yalnızca Agent/Chat'te geçerli)

8.6. Inline Edit Kullanım Senaryoları

Senaryo	Örnek Talimat	Neden Inline Edit?
Fonksiyon refactoring	"async/await'e çevir"	Tek fonksiyon, tek dosya
Değişken yeniden adlandırma	"camelCase kullan"	Cerrahi, hızlı
Hata yönetimi ekleme	"try-catch ekle"	Yerel kapsam
Bileşen dönüşümü	"class'tan functional component'e çevir"	Tek dosya
Kod açıklama	Soru modu: "Bu regex ne yapıyor?"	Chat açmaya gerek yok

9. Agent Orkestrasyonu ve Model Harness

Cursor Agent, frontier model + yerleşik araçlar + Rules ile çalışır. **Agent harness**, modele verilen talimatlar, araç şemaları ve bağlam mühendisliği paketidir.

9.1. Harness bileşenleri

Bileşen	İçerik	Model bağımlılığı
System prompt	Persona, tool-use kuralları	Her model için ayrı optimize
Tool schemas	OpenAI / Anthropic / Gemini format	Format dönüşümü gerekir
Context budget	Token önceliklendirme	Model penceresine göre
Compaction policy	Özetleme tetikleyicisi	Ortak harness

Dynamic discovery	Dosya tabanlı context	Ortak harness
-------------------	-----------------------	---------------

Resmi blog: "Cursor's agent harness is optimized individually for every new frontier model we support."

9.2. Tool calling format uyumluluğu

Sağlayıcı	Format	Cursor adaptasyonu
Anthropic	XML tool_use blocks	Native API
OpenAI	function_call / tools	Schema mapping
Google Gemini	function calling	Gemini-specific
Cursor Composer	Proprietary	Native optimizasyon

9.3. Agent modları (Classic IDE)

Mod	Yazma	Kullanım
Ask	Hayır	Anlama
Edit	Evet	Hedefli değişiklik
Agent	Evet	Otonom görev
Plan	Plan aşamasında hayır	Shift+Tab
Debug	Evet	Hipotez-log-fix

9.4. Orkestrasyon döngüsü

```
graph TD
    START["Kullanıcı prompt"] --> RULES["Rules enjekte"]
    RULES --> CTX["Bağlam derle + dynamic paths"]
    CTX --> LLM["Model inference"]
    LLM --> TC{"Tool call?"}
    TC -->|Evet| APPROVE["Onay / Run Mode"]
    APPROVE --> EXEC["İstemci yürüt"]
    EXEC --> CTX
    TC -->|Hayır| OUT["Yanıt + diff"]
```

Checkpoint'ler Git'ten bağımsız yerel snapshot'tır. Queue (Enter) vs Steer (Cmd+Enter) mesaj enjeksiyonu desteklenir.

9.5. Agent operasyonel detayları

10. Agent Araçları: Instant Grep vs ripgrep

Agent kod tabanında metin ararken varsayılan olarak **Instant Grep** kullanır; bu, Mart 2026 Cursor blog'unda detaylandırılan trigram inverted index tabanlı özel motordur.

10.1. Karşılaştırma

Özellik	ripgrep (rg)	Instant Grep
Algoritma	Lineer dosya taraması	Trigram inverted index
Monorepo 50k+ dosya	15+ sn [GÖZLEM]	Milisaniye-saniye [resmi hedef]
Regex desteği	Tam	Tam + kelime sınırı
İndeks	Yok (her seferinde okuma)	Arka planda sürdürülen
Yapılandırma	Agent otomatik seçer	Otomatik, kullanıcı ayarı yok

10.2. Trigram indeksleme (Resmi özet)

- Her dosyadan overlapping 3-karakter token'lar çıkar
- Inverted index: trigram → posting list (dosya ID'leri)
- Regex sorgusu trigram ağacına ayrıştırılır
- Posting list kesişimi aday dosyaları daraltır
- Yalnızca adaylarda tam regex doğrulama

```
graph LR
  Q["Regex sorgusu"] --> DEC["Trigram ayrıştırma"]
  DEC --> IDX["Inverted Index"]
  IDX --> CAND["Aday dosyalar"]
  CAND --> VERIFY["Tam eşleşme"]
  VERIFY --> AGENT["Agent bağlamı"]
```

10.3. Instant Grep vs @codebase (semantik)

	Instant Grep	@codebase
Eşleştirme	Sözdizimsel / regex	Embedding benzerliği
Sorgu	Bilinen sembol, hata string'i	"Auth nasıl çalışır?"
Backend	Yerel indeks	Turbopuffer

10.4. Explore Subagent

Geniş arama görevlerinde ana bağlamı korumak için paralel alt ajan; yalnızca özet bulguları ana konuşmaya döner.

11. Shadow Workspace ve gRPC IPC

Shadow Workspace, AI'nın yazdığı kodu **gizli Electron penceresinde** LSP ile doğrulamasını sağlar (resmi Cursor blog).

11.1. Tasarım kriterleri

Kriter	Açıklama
--------	----------

LSP-usability	Gerçek linter/definition
Runnability	Kod çalıştırılabilir olmalı (kısmi)
Independence	Kullanıcı UI'ını bloklamaz
Privacy	Gizli pencere
Concurrency	İstekler arası pencere yeniden kullanımı

11.2. gRPC proto (reconstructed) [TERS MÜHENDİSLİK / GÖZLEM]

Resmi blog gRPC + buf kullanıldığını belirtir; tam `.proto` public değildir. Gözlemlenen mesaj yapısı:

```
// shadow_workspace.v1 – [TERS MÜHENDİSLİK, yapısal rekonstrüksiyon]
syntax = "proto3";
package shadow_workspace.v1;

service ShadowWorkspaceService {
  rpc ApplyEdit(ApplyEditRequest) returns (ApplyEditResponse);
  rpc GetDiagnostics(GetDiagnosticsRequest) returns (GetDiagnosticsResponse);
  rpc SyncWorkspace(SyncWorkspaceRequest) returns (SyncWorkspaceResponse);
}

message ApplyEditRequest {
  string workspace_id = 1;
  string relative_path = 2;
  string new_contents = 3;
  int32 version = 4;
}

message GetDiagnosticsResponse {
  repeated DiagnosticItem items = 1;
  int32 error_count = 2;
  int32 warning_count = 3;
}

message DiagnosticItem {
  int32 line = 1;
  int32 column = 2;
  string message = 3;
  string severity = 4; // error | warning | info
}
```

11.3. IPC akışı

Extension Host (görünür) ↔ Extension Host (shadow) arası gRPC; renderer'lar doğrudan konuşmaz.

11.4. Sınırlamalar [resmi blog *]

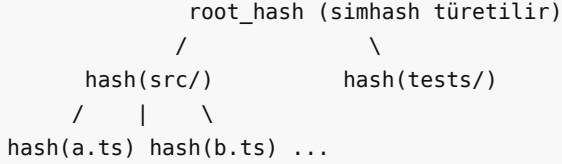
Tam disk izolasyonu (runnability) henüz ideal değil; kernel-level folder proxy gelecek hedef olarak belirtilmiştir.

12. Kod Tabanı İndeksleme: Merkle, Simhash, Content Proof

Semantik arama (@codebase), Agent performansını ~%12.5 artırır (resmi değerlendirme). Pipeline Classic IDE açılışında arka planda çalışır.

12.1. Merkle ağacı

Her dosya SHA-256 hash; klasör hash'i çocuk hash'lerinden türetilir (Git benzeri).



Dosya değişince yalnızca yaprak → kök yolu yeniden hesaplanır. 50.000 dosyada ~3.2 MB metadata; ağaç sayesinde yalnızca farklı dallar senkronize edilir.

12.2. Simhash ve indeks paylaşımı

Yeni kullanıcı açılışında:

1. İstemci Merkle ağacı + **simhash** hesaplar
2. Simhash sunucuya yüklenir
3. Takım içi vektör DB'de benzer simhash aranır
4. Eşik üstü eşleşme (~%92 benzerlik org ortalaması) → **indeks kopyalama**
5. Turbopuffer `copy_from_namespace` [GÖZLEM]

Percentil	İlk sorgu (naif)	Simhash reuse sonrası
Medyan	7.87 sn	525 ms
P90	2.82 dk	1.87 sn
P99	4.03 saat	21 sn

12.3. Content proof güvenliği

Kopyalanan indeksten sorgularken istemci **tam Merkle ağacını** kanıt olarak yükler. Sunucu sonuçları filtreler: istemci hash'i kanıtlamadığı dosya sonuçlardan düşer. Kök hash eşleşince kanıtlar silinir.

```
sequenceDiagram
    participant C as İstemci
    participant S as Cursor Sunucu
    participant TP as Turbopuffer

    C->>S: simhash + Merkle tree (proof)
    S->>TP: kopyalanmış namespace sorgusu
    TP-->>S: aday chunk'lar
    S->>S: content proof filtresi
    S-->>C: obfuscated path + satır aralığı
    C->>C: yerel dosya oku → LLM bağlamı
```

Kritik: Kaynak kod plaintext olarak sunucuda saklanmaz; yalnızca embedding + obfuscated metadata.

12.4. Chunking ve embedding cache

Dosyalar sözdizimsel (AST-aware) parçalara bölünür. Embedding'ler **içerik hash'ine** göre cache'lenir; değişmeyen chunk yeniden embed edilmez.

12.5. Yapılandırma

Dosya	Etki
.cursorignore	AI erişimini engeller (terminal/MCP bypass edebilir)
.cursorindexingignore	Yalnızca indeksleme hariç
.cursor/keys	Özel path decryption key

12.6. İndeksleme operasyonel detayları

13. Turbopuffer Vektör Deposu

Turbopuffer, Cursor'ın semantik indeks backend'idir: namespace-per-codebase modeli, object storage + NVMe hibriti.

13.1. Namespace modeli [GÖZLEM]

Kavram	Açıklama
Namespace	Genellikle codebase/workspace başına bir
Aktif namespace	Bellek/NVMe'de sıcak
İnaktif namespace	Object storage'a soğur
Warm-up	Sorgu gelince namespace ısıtılır
copy_from_namespace	Takım indeks paylaşımı

13.2. Saklanan veri

Veri türü	Sunucuda?
Embedding vektörü	Evet
Obfuscated path	Evet
Chunk içeriği (plaintext)	Hayır
Satır aralığı metadata	Evet (şifreli/obfuscated)

13.3. Sorgu akışı

- Sorgu metni → embedding (istemci veya sunucu)
- Turbopuffer ANN (approximate nearest neighbor)
- Content proof filtresi

4. İstemci yerel okuma → LLM

13.4. Ölçek [GÖZLEM – üçüncü taraf analiz]

Turbopuffer üzerinde trilyonlarca vektör ve on milyonlarca namespace raporlanmıştır; Cursor en büyük tüketicilerden biridir.

14. Dynamic Context Discovery

Statik bağlam enjeksiyonu yerine agent'ın ihtiyaç duyduğu veriyi **dosya sisteminden dinamik çekmesi** (resmi blog, Ocak 2026).

14.1. Dynamic vs static

Static context	Dynamic context
Tüm MCP tool tanımları prompt'ta	MCP tools → klasör senkronizasyonu
Uzun shell çıktısı kesilir	Çıktı dosyaya yazılır, agent tail/read
Tüm Skills tam metin	Skill adı static; içerik grep ile
Terminal geçmişi yapıştırma	Terminal session → dosya

14.2. Dynamic context path'leri [GÖZLEM]

Path türü	Amaç
Tool output spill	Büyük JSON/shell çıktısı
Chat history file	Compaction sonrası arama
MCP tool sync folder	Sunucu başına alt klasör
Terminal session sync	Entegre terminal log
Skill/Rules dosyaları	@mention veya grep ile

MCP A/B testi: MCP kullanan agent run'larında toplam token **%46.9 azalma** (istatistiksel anlamlı).

14.3. Akış

```
graph TD
    TC["Tool call büyük yanıt"] --> FILE["Spill → yerel dosya"]
    FILE --> AG["Agent read/tail/rg"]
    AG --> CTX["Minimal özet bağlama"]
    MCP["MCP sunucuları"] --> SYNC["Tool desc sync klasörü"]
    SYNC --> AG
```

Harness tüm modellerde ortak; discovery pattern modelden bağımsızdır.

15. Context Ring, Compaction ve preCompact Hook

Agent oturumları **context ring** (dönüşümlü bağlam penceresi) ile yönetilir: mesajlar, tool sonuçları ve sistem prompt token bütçesini doldurur.

15.1. Context ring yapısı [GÖZLEM]

Slot türü	Öncelik	Compaction davranışı
System prompt + Rules	Sabit yüksek	Korunur / sıkıştırılmaz
Son N tool sonucu	Orta	Özetlenir veya dosyaya spill
Eski konuşma	Düşük	Summarization
Dynamic file refs	Talep üzerine	Dosyadan yeniden okunabilir

Composer **self-summarization** eğitimi ile compaction kalitesi artırılır: model kendi bağlamını özetler, plan durumu ve kalan görevler korunur.

15.2. Compaction tetikleyicileri

Tetikleyici	Eşik [GÖZLEM]
Otomatik	~%70–85 context doluluğu (model değişken)
Manuel	/summarize komutu
preCompact hook	Compaction hemen öncesi

15.3. preCompact hook (Resmi docs)

Gözlemsel hook — compaction'ı bloklayamaz veya değiştiremez.

`.cursor/hooks.json` örneği:

```
{
  "version": 1,
  "hooks": {
    "preCompact": [
      { "command": "./scripts/on-pre-compact.sh" }
    ]
  }
}
```

Girdi payload (stdin JSON):

Alan	Tip	Açıklama
trigger	string	auto manual
context_usage_percent	number	Doluluk yüzdesi
context_tokens	number	Kullanılan token
context_window_size	number	Model penceresi

message_count	number	Toplam mesaj
messages_to_compact	number	Özetlenecek mesaj
is_first_compaction	boolean	İlk compaction mı
conversation_id	string	Oturum UUID

Çıktı: Opsiyonel `user_message` (kullanıcıya bildirim). `followup_message` desteklenmez.

15.4. preCompact vs Claude Code PreCompact

	Cursor preCompact	Claude Code PreCompact
Bloklama	Hayır	Evet (bekletme)
Özet metni hook'a	Hayır	Kısmen
Kullanım	Telemetry, transcript snapshot	Zorunlu kayıt

Workaround: hook içinde senkron transcript mining (115s bütçe [GÖZLEM]).

16. Rules, Skills, Hooks ve MCP

16.1. Rules önceliği

```
graph TD
    TR["Team Rules"] --> MERGE["Birleştirilmiş set"]
    PR["Project .mdc"] --> MERGE
    UR["User Rules"] --> MERGE
    AG["AGENTS.md"] --> MERGE
    MERGE --> PROMPT["System prompt"]
```

Tür	Konum	Tab/Cmd+K
Team Rules	Dashboard	Hayır
Project Rules	<code>.cursor/rules/*.mdc</code>	Hayır
User Rules	Settings	Hayır
AGENTS.md	Proje kökü	Hayır

`.mdc` frontmatter: `alwaysApply` , `description` , `globs` .

16.2. Skills

`SKILL.md` + frontmatter; dynamic discovery ile yüklenir (Bölüm 14).

16.3. Hooks özeti

Hook	Bloklayabilir?
------	----------------

beforeShellExecution	Evet
beforeMCPExecution	Evet
beforeReadFile	Evet
beforeSubmitPrompt	Evet
preCompact	Hayır (observe)
stop	Evet (exit code)
beforeTabFileRead	Evet

16.4. MCP

Transport: stdio, SSE, Streamable HTTP. Proje: `.cursor/mcp.json` , global: `~/.cursor/mcp.json` .

Interpolation: `${env:NAME}` , `${workspaceFolder}` , `${userHome}` .

Enterprise: MCP Allowlist, ağ politikası (allow all / allowlist / deny all).

16. Rules: Kalıcı Yapay Zeka Talimatları

16.1. Rules Mimarisi

Rules, Agent'a sistem düzeyinde kalıcı talimatlar sağlar. LLM'ler tamamlamalar arasında bellek tutmadığı için Rules, **prompt seviyesinde kalıcı bağlam** işlevi görür.

```
graph TD
    REQ["Kullanıcı İsteği"] --> MERGE["Kural Birleştirme"]
    TR["Team Rules<br/>(En Yüksek Öncelik)"] --> MERGE
    PR["Project Rules<br/>(.cursor/rules/*.mdc)"] --> MERGE
    UR["User Rules<br/>(Global)"] --> MERGE
    AG["AGENTS.md<br/>(Alternatif)"] --> MERGE
    MERGE --> CTX["Model Bağlamına Enjeksiyon"]
    CTX --> AGENT["Agent Yanıtı"]
```

16.2. Kural Türleri

Tür	Konum	Kapsam	Versiyon Kontrolü
Team Rules	Cursor Dashboard	Tüm takım, tüm repo	Dashboard yönetimi
Project Rules	<code>.cursor/rules/*.mdc</code>	Proje	Git ile
User Rules	Cursor Settings → Customize → Rules	Tüm projeler	Cursor hesabı
AGENTS.md	Proje kökü / alt dizinler	Proje/dizin	Git ile
.cursorsrules	Proje kökü	Proje (legacy)	Git ile — deprecated

16.3. Project Rules (.mdc) Formatı

Her kural `.mdc` uzantılı markdown dosyasıdır; YAML frontmatter içerir:

```
---
description: "RPC servis konvansiyonları"
globs: src/services/**/*.ts
alwaysApply: false
---
```

- Her servis kendi dosyasında tanımlanmalı
- Girdi doğrulaması servis sınırında yapılmalı
- Yapılandırılmış hata nesneleri dön (code + message)

Kural Uygulama Türleri

Kural Türü	Frontmatter	Davranış
Always Apply	alwaysApply: true	Her chat oturumunda dahil
Apply Intelligently	description dolu	Agent açıklamaya göre karar verir
Apply to Specific Files	globs dolu	Eşleşen dosya bağlamdaysa dahil
Apply Manually	@-mention	Kullanıcı @rule-name ile çağırır

Frontmatter Etkileşim Tablosu

alwaysApply	description	globs	Sonuç
true	—	—	Her zaman dahil
false	—	pattern	Eşleşen dosya bağlamdaysa otomatik
false	dolu	—	Agent relevance kararı
false	—	—	Yalnızca @-mention ile

16.4. AGENTS.md

.cursor/rules yerine basit markdown alternatifi:

```
# Proje Talimatları

## Kod Stili
- Tüm yeni dosyalar TypeScript olmalı
- React'te functional component tercih et
- Veritabanı sütunları snake_case

## Mimari
- Repository pattern kullan
- İş mantığını service katmanında tut
```

Nested AGENTS.md: Alt dizinlerdeki AGENTS.md dosyaları, o dizindeki dosyalarla çalışırken otomatik uygulanır. Daha spesifik talimatlar üst dizin talimatlarını override eder.

16.5. Team Rules (Kurumsal)

Team/Enterprise planlarda dashboard üzerinden yönetilir:

- **Enable immediately:** Oluşturulunca aktif
- **Enforce:** Tüm takım üyeleri için zorunlu, kapatılamaz
- **Glob patterns:** Dosya kapsamı sınırlama desteklenir
- **Öncelik:** Team Rules > Project Rules > User Rules

16.6. Rules Oluşturma Yöntemleri

1. **Chat'te** `/create-rule` — Agent'a kural oluşturmasını söyleyin
2. **Customize** → **Rules** → **Add Rule** — GUI üzerinden
3. **Remote Rules (GitHub)** — Başka repodan `.mdc` import
4. **Manuel dosya oluşturma** — `.cursor/rules/` altına `.mdc` ekleyin

16.7. Rules En İyi Uygulamalar

Yapılması gerekenler:

- Kuralları 500 satırın altında tutun
- Büyük kuralları parçalayın
- Somut örnekler ve @file referansları kullanın
- Git'e commit edin (takım paylaşımı)
- Agent aynı hatayı tekrarlırsa kural ekleyin

Yapılmaması gerekenler:

- Tüm stil rehberini kopyalamayın (linter kullanın)
- Her olası komutu belgelemeyin (Agent zaten biliyor)
- Nadiren uygulanan edge case'ler için kural eklemeyin
- Kod tabanındaki kodu kopyalamayın (@file referans verin)

16.8. Rules'ın Etki Alanı

Özellik	Rules Etkiler mi?
Agent (Chat)	Evet
Plan Mode	Evet
Inline Edit (Cmd+K)	Hayır
Tab	Hayır
User Rules → Inline Edit	Hayır

16.5. MCP operasyonel detayları

17. Agent Sandbox: Seatbelt ve Landlock

Cursor Agent terminal komutlarını platform-native sandbox ile izole eder (resmi blog, 2026).

17.1. Platform matrisi

Platform	Mekanizma	Durum
----------	-----------	-------

macOS	Seatbelt (sandbox-exec)	Shipped
Linux	Landlock + seccomp + overlay FS	Shipped
Windows	WSL2 içinde Linux sandbox	Shipped
Windows native	Microsoft iş birliği	Gelecek

17.2. macOS Seatbelt [GÖZLEM / resmi]

Dinamik .sb profili runtime'da üretilir:

Kaynak	Policy etkisi
Workspace kökü	read/write allow
.cursorignore	deny read/write
Admin ayarları	Ek kısıtlar
Ağ	Kısıtlı (GitHub, web search allowlist)

Örnek profil parçası [TERS MÜHENDİSLİK – yapısal] :

```
(version 1)
(deny default)
(allow file-read* (subpath "/workspace/project"))
(allow file-write* (subpath "/workspace/project"))
(deny file-read* (regex "^.*\\.env$"))
(allow network-outbound (remote tcp "api.github.com:443"))
```

Seatbelt 2016'da deprecated ilan edilmiştir ancak Chrome vb. hâlâ kullanır.

17.3. Linux Landlock + seccomp

Katman	İşlev
seccomp-bpf	Tehlikeli syscall engeli
Landlock	FS erişim kuralları
Overlay FS	.cursorignore dosyaları Landlocked kopya ile üzerine yazılır

Init gecikmesi: overlay remount en yavaş adım (path bilgisi seccomp'ta zor).

17.4. Run Modes ile ilişki

Sandbox, Run Mode allowlist'ten bağımsız ikinci savunma hattıdır. Allowlist otomatik onay; sandbox FS/ağ sınırı uygular.

18. BYOK, Kimlik Doğrulama ve state.vscdb

Kimlik bilgileri VS Code uyumlu SQLite veritabanında saklanır.

18.1. Dosya konumları

Platform	state.vscdb yolu
macOS	~/Library/Application Support/Cursor/User/globalStorage/state.vscdb
Linux	~/.config/Cursor/User/globalStorage/state.vscdb
Windows	%APPDATA%\Cursor\User\globalStorage\state.vscdb

18.2. ItemTable anahtarları [TERS MÜHENDİSLİK]

```
-- Şema (VS Code globalStorage)
CREATE TABLE ItemTable (key TEXT PRIMARY KEY, value TEXT);
```

Key	İçerik	BYOK?
cursorAuth/accessToken	JWT access token	Hayır
cursorAuth/refreshToken	JWT refresh	Hayır
cursorAuth/token	Alternatif token key	Hayır
storage.serviceMachineId	Machine ID (checksum)	Hayır
secret://{provider_id}	safeStorage şifreli API key	Evet
cursorAuth/cachedEmail	Hesap email	Hayır
cursorAuth/stripeMembershipType	Plan türü	Hayır
cursorai/featureConfig	Feature flags JSON	Hayır
workbench.panel.aichat.view.aichat.chatdata	Chat oturum metadata	Hayır

18.3. BYOK yolu

1. Cursor Settings → Models → API Keys
2. Key Electron safeStorage ile şifrelenir
3. ItemTable 'a secret:// prefix ile yazılır
4. Inference isteğinde ModelDetails.api_key veya provider-specific alan protobuf'a enjekte edilir

```
message ModelDetails {
  optional string model_name = 1;
  optional string api_key = 2;
  optional AzureState azure_state = 4;
  optional BedrockState bedrock_state = 7;
  optional string openai_api_base_url = 6;
}
```

BYOK aktifken istek doğrudan sağlayıcıya veya Cursor proxy üzerinden gidebilir; x-ghost-mode Privacy Mode ile etkileşir.

18.4. Checksum header [TERS MÜHENDİSLİK]

`x-cursor-checksum = jyh_cipher(timestamp) + machine_id` — sunucu tarafı istemci bütünlüğü kontrolü.

19. Composer RL ve 5 Saatlik Üretim Döngüsü

Composer, Cursor'ın agentic coding için RL ile eğittiği özel MoE modeldir. **Real-time RL** üretim etkileşimlerinden ödül sinyali türetir.

19.1. Döngü (Resmi — ~5 saat)

```
graph LR
    D["Deploy checkpoint"] --> C["Collect milyarlarca token"]
    C --> R["Reward distillation"]
    R --> T["Train weights"]
    T --> E["CursorBench eval"]
    E -->|pass| D
    E -->|fail| X["Rollback"]
```

Adım	Süre [resmi]	Detay
Toplam döngü	~5 saat	Günde birden çok checkpoint
Veri	On-policy	Deploy edilen model = veri üreten model
Eval	CursorBench	Regresyon kapısı
Deploy hedefi	Auto router	İyileşmiş Composer

19.2. Reward sinyalleri [GÖZLEM / resmi özet]

Sinyal	Yorum
Edit retained	Kullanıcı değişikliği korudu
Follow-up dissatisfaction	Olumsuz devam mesajı
Task completion	Görev başarılı kapandı
Tool call outcome	Terminal/test başarısı

Tab'daki basit $\pm$ ödülün aksine agent reward çok boyutlu ve gürültülüdür; büyük batch gerekir.

19.3. Harness parity

Composer eğitimi **production harness ile aynı** tool set, prompt format ve semantic search davranışı ile yapılır. Shadow backend deployment eğitim sırasında production'ı yansıtır.

19.4. Self-summarization RL

Uzun horizon görevlerde model compaction sırasında kendi bağlamını özetlemeyi öğrenir; kötü özetler final reward ile cezalandırılır.

20. Bağlam Derleme ve Model Yönlendirme

20.1. Bağlam kaynakları

Kaynak	Tab	Cmd+K	Agent
Açık dosyalar	Evet	Evet	Evet
@codebase	Hayır	Hayır	Evet
Rules	Hayır	Hayır	Evet
MCP	Hayır	Hayır	Evet
Terminal dosyası	Hayır	Hayır	Evet (dynamic)
LSP diagnostics	Evet	Evet	Evet

20.2. Model router (2026)

Model	Sağlayıcı	Varsayılan ctx	Max ctx
Claude 4.6/4.7 Opus	Anthropic	200–300k	1M
Claude Sonnet 5	Anthropic	200k	1M
Composer 2.5	Cursor	200k	—
GPT-5.5	OpenAI	272k	1M
Gemini 3.7 Flash	Google	200k	1M
Grok 4.6	Cursor/xAI	256k	—

Auto mode: Cursor Router görev tipine göre model seçer (planning → güçlü model, apply → Composer).

20.3. Token bütçe önceliklendirme

1. System + Rules (yüksek)
2. Aktif edit hedefi (yüksek)
3. Son tool sonuçları (orta)
4. @mention dosyalar (orta-düşük)
5. Eski konuşma (düşük → compaction)

20.4. Bağlam derleme genişletmesi

21. Gizlilik, Güvenlik ve Veri Yönetimi

21.1. Veri Akışı Modeli

Cursor Classic IDE'de veri üç katmanda hareket eder:

1. **Tamamen yerel:** Dosya okuma/yazma, Tab önerileri (kısmi), checkpoint'ler, git operasyonları
2. **Şifreli bulut iletişimi:** Embedding vektörleri, model inference istekleri, obfuscated metadata
3. **Harici sistemler:** MCP sunucuları, GitHub, web arama sağlayıcıları

21.2. İndeksleme Gizliliği

Veri	Sunucuda Saklanır mı?	Açıklama
Kaynak kod (plaintext)	Hayır	Yalnızca istemci tarafında okunur
Embedding vektörleri	Evet	Semantik arama için
Dosya yolları	Gizlenmiş/şifreli	Obfuscation + opsiyonel custom key
Kod parçaları	Şifreli, geçici	İndeksleme sırasında bellekte, sonra atılır

Dosya yolu şifreleme: `.cursor/keys` ile özel anahtar tanımlanabilir.

Takım paylaşımı: İndeksler takım üyeleri arasında paylaşılabilir; dosya erişim izinleri korunur.

21.3. Privacy Mode

Privacy Mode etkinleştirildiğinde:

- Bulut indekslemesi devre dışı kalır veya kısıtlanır
- Embedding'ler yerelde tutulur
- Kod içeriğinin Cursor sunucularına gönderilmesi minimize edilir
- Kurumsal compliance gereksinimleri için kritik

Enterprise müşterileri için ek veri saklama politikaları ve admin kontrolleri dashboard üzerinden yönetilebilir.

21.4. Agent Güvenlik Guardrail'leri

Eylem	Varsayılan Davranış
Dosya okuma	Onaysız
Kod tabanı arama	Onaysız
Workspace dosya düzenleme	Onaysız (config dosyaları hariç)
Config dosyası düzenleme	Onay gerekli
Terminal komutu	Onay gerekli
MCP tool call	Onay gerekli (allowlist hariç)
Ağ isteği	Yalnızca GitHub, link fetch, web search

Run Modes: Allowlist tabanlı otomatik onay veya Auto-review sınıflandırıcı.

Workspace Trust: VS Code workspace trust desteklenir; varsayılan kapalı. Etkinleştirildiğinde güvenilmeyen workspace'lerde kısıtlı mod.

21.5. .cursorignore ve Güvenlik

`.cursorignore` AI erişimini kısıtlar ancak **mutlak koruma garantisi vermez** — LLM öngörülemezliği nedeniyle. Hassas dosyalar için:

- `.cursorignore` + `.gitignore` birlikte kullanın
- Global ignore patterns tanımlayın (`**/.env` , `**/*.pem` , `**/secrets.json`)

- MCP sunucularının erişim kapsamını sınırlayın
- Terminal komutları için Run Mode'u sıkı tutun

21.6. Prompt Injection ve Sorumlu Açıklama

Cursor, prompt injection, halüsinasyon ve beklenmedik model davranışlarına karşı guardrail'ler uygular.

Güvenlik açığı bildirimleri: security-reports@cursor.com — 5 iş günü içinde onay yanıtı.

21.7. Model Veri Saklama

Model sağlayıcılarına göre veri saklama politikaları değişir. Bazı modeller (ör. Claude Fable 5) Enterprise müşterileri için ek veri saklama onayı gerektirir. Cursor dashboard'dan model seçimi ve veri politikası bilgileri görüntülenebilir.

22. Git Entegrasyonu ve Sürüm Kontrolü

22.1. AI Destekli Git Özellikleri

Cursor Classic IDE, VS Code Source Control panelinin üzerine AI destekli git özellikleri ekler:

```
graph LR
  subgraph "Git Entegrasyonu"
    CM["AI Commit Message<br/>🌟 Sparkle Icon"]
    MC["Merge Conflict<br/>Resolve in Chat"]
    ATTR["Attribution<br/>Made with Cursor"]
    BLAME["Cursor Blame<br/>AI vs Human"]
  end

  DEV["Geliştirici"] --> CM
  DEV --> MC
  AGENT["Agent"] --> ATTR
  AGENT --> BLAME
```

22.2. AI Commit Mesajları

1. Değişiklikleri stage edin
2. Commit mesajı alanındaki **sparkle (🌟) ikonuna** tıklayın
3. Cursor diff + repo geçmişine dayalı mesaj üretir

22.3. Merge Conflict Çözümü

Merge conflict durumunda **"Resolve in Chat"** butonu:

1. Agent her iki tarafı analiz eder
2. Birleştirme önerisi sunar
3. Geliştirici diff'i inceler ve kabul eder

22.4. Agent Attribution

Agent tarafından oluşturulan commit ve PR'lara otomatik `Made with Cursor` trailer eklenebilir:

- **Varsayılan:** Açık
- **Ayarlama:** Cursor Settings → Agent → Attribution (3.11+ → Git & PRs → Attribution)
- **Enterprise override:** Admin Dashboard'dan global kapatma

22.5. Cursor Blame

Cursor Blame, git geçmişinde hangi satırların AI, hangilerinin insan tarafından yazıldığını gösterir:

- Code review süreçlerinde AI footprint analizi
- Denetim (audit) ve compliance
- Kod tabanında AI kullanım oranının anlaşılması

22.6. Agent ve Git Workflow

Agent Classic IDE'de git operasyonlarını terminal aracılığıyla yürütür:

- `git status`, `git diff`, `git log` analizi
- Branch oluşturma ve commit
- `gh pr create` ile PR oluşturma
- Merge conflict analizi

Paralel Agent'lar git worktree kullanarak izole branch'lerde çalışabilir.

22.7. Checkpoint vs Git

Özellik	Checkpoint	Git
Kapsam	Agent oturumu	Kalıcı versiyon kontrolü
Depolama	Yerel, geçici	.git dizini
Geri alma	Chat timeline	<code>git revert/reset</code>
Takım paylaşımı	Hayır	Evet
Kullanım	Agent deneme/yanlış yön	Kalıcı kayıt

23. VS Code Uyumluluğu ve Geçiş

23.1. Fork İlişkisi

Cursor, VS Code kod tabanına dayanır. Bu stratejik karar şu avantajları sağlar:

- Tanıdık editör deneyimi (sekmeler, split view, minimap)
- VS Code eklenti ekosistemi uyumluluğu
- LSP, DAP, TextMate grammar desteği
- VS Code Settings API
- Keyboard shortcuts ve Command Palette

Cursor düzenli olarak VS Code'un güncel sürümüne **rebase** edilir; ancak stabilite için genellikle biraz eski VS Code sürümü kullanılır.

23.2. VS Code'dan Geçiş

Tek Tıkla Import

1. Cursor Settings (`Cmd+Shift+J` / `Ctrl+Shift+J`)
2. General → Account → "VS Code Import" → Import
3. Aktarılan: Extensions, Themes, Settings, Keybindings

Profil Migrasyonu (Manuel)

Export (VS Code):

1. Command Palette → "Preferences: Open Profiles (UI)"
2. Profil → 3-nokta → Export Profile → local file veya GitHub Gist

Import (Cursor):

1. Command Palette → "Preferences: Open Profiles (UI)"
2. New Profile yanındaki ok → Import Profile
3. Gist URL veya dosya seç → Import → Profili aktifleştir

Terminal ile Eklenti Kopyalama

```
cp -r ~/.vscode/extensions/* ~/.cursor/extensions/
```

Dikkat: Cursor **Open VSX** registry kullanır, VS Code Marketplace değil. Microsoft eklentileri (C/C++, Pylance, Remote-*) kopyalansa bile çalışmayabilir; Extensions sidebar'dan yeniden kurulum önerilir.

23.3. Yapılandırma Dosya Konumları

Platform	Cursor Config	Cursor Extensions	VS Code Config
macOS	~/Library/Application Support/Cursor	~/.cursor/extensions	~/Library/Application Support/Code
Linux	~/.config/Cursor	~/.cursor/extensions	~/.config/Code
Windows	%APPDATA%\Cursor	%USERPROFILE%\cursor\extensions	%APPDATA%\Code

Cursor ve VS Code **ayrı dizinler** kullanır; paralel çalıştırma mümkündür. Ancak aynı proje iki editörde açıksa dosya kilitleme/otomatik formatlama çakışmaları olabilir.

23.4. Ayar Menüleri

Menü	Erişim
Cursor Settings	Command Palette → "Cursor Settings"
VS Code Settings	Command Palette → "Preferences: Open Settings (UI)"

23.5. Activity Bar Yönelimi

Cursor varsayılan olarak yatay activity bar kullanır (AI chat için alan optimizasyonu). Dikey'e dönmek için:

1. Settings → `workbench.activityBar.orientation` → `vertical`
2. Cursor'ı yeniden başlatın

23.6. Bilinen Uyumluluk Sınırlamaları

Konu	Durum
VS Code Marketplace eklentileri	Open VSX üzerinden; bazı MS eklentileri kısıtlı
VS Code Settings Sync	Cursor'da native cloud sync yok; profil export/import kullanın

Remote Development	Cursor'ın kendi remote çözümü; VS Code Remote sınırlı
Dev Containers	Desteklenir (VS Code tabanlı)
Import Settings butonu	Bilinen bug: eklentileri atlayabilir; profil export önerilir

23.7. Classic IDE Terminal Başlatma

```
# Classic IDE modunda proje aç
cursor . --classic

# Classic olmadan (Agents Window açabilir)
cursor .

# Belirli dosya
cursor --classic path/to/file.ts
```

24. Performans Profili ve Kapasite Planlama

Bileşen	CPU	RAM	Disk I/O	Ağ
Tab Engine	Düşük-Orta	200-500 MB	Minimal	Sürekli düşük
Instant Grep indeks	Orta	500 MB-1 GB	Orta	Yok
İlk indeksleme	Yüksek	500 MB-2 GB	Yüksek	Artımlı upload
Agent (aktif)	Orta-Yüksek	1-4 GB	Orta	Yüksek
Shadow Workspace	Orta	+200-500 MB	Orta	Yok
MCP sunucuları	Değişken	Değişken	Değişken	Değişken
Extension Host	Değişken	200-1000 MB	Düşük	LSP

Monorepo: `.cursorindexingignore` ile generated/vendor hariç tutun. İlk indeksleme sonrası Merkle artımlı sync hafiftir.

24. Bileşen Karşılaştırma Tabloları

24.1. AI Özellik Karşılaştırması

Özellik	Tab	Inline Edit (Cmd+K)	Agent (Cmd+I/L)
Gecikme	~320 ms	~1-3 sn	~5-60+ sn
Kapsam	Mevcut dosya (+ cross-file portal)	Seçili kod, tek dosya	Çok dosya, terminal, web, MCP
Model	~8B özel model	Frontier LLM	Frontier LLM (seçilebilir)

Rules etkisi	Hayır	Hayır	Evet
Terminal erişimi	Hayır	Hayır	Evet
Onay gereksinimi	Hayır	Diff onayı	Tool call onayları
Checkpoint	Hayır	Hayır	Evet
Plan Mode	Hayır	Hayır	Evet
MCP erişimi	Hayır	Hayır	Evet

24.2. Klavye Kısayolları (Classic IDE)

Eylem	macOS	Windows/Linux
Agent aç	Cmd+I	Ctrl+I
Chat aç	Cmd+L	Ctrl+L
Inline Edit	Cmd+K	Ctrl+K
Inline Edit soru modu	Opt+Return	Alt+Return
Tab kabul	Tab	Tab
Tab kelime kabul	Cmd+↔	Ctrl+↔
Editor Window aç	Shift+Cmd+N	Shift+Ctrl+N
Plan Mode toggle	Shift+Tab	Shift+Tab
Cursor Settings	Cmd+Shift+J	Ctrl+Shift+J
Command Palette	Cmd+Shift+P	Ctrl+Shift+P
Kuyruğa mesaj	Enter	Enter
Anında mesaj (steer)	Cmd+Enter	Ctrl+Enter

24.3. Bağlam Penceresi Boyutları

Model	Varsayılan	Maksimum	Notlar
Claude 4.6/4.7 Opus	200-300k token	1M	Max Mode gerekli (legacy planlar)
Claude Sonnet 5	200k	1M	Geniş bağlam, güncellenmiş tokenizer
Composer 2.5	200k	—	Cursor özel model, hız odaklı
GPT-5.5	272k	1M	Uzun bağlam 2x input fiyat
Gemini 3.7 Flash	200k	1M	Monorepo analizi
Grok 4.6	256k	—	Cursor + SpaceXAI ortak eğitim

24.4. Rules vs Skills vs AGENTS.md

Boyut	Rules	Skills	AGENTS.md
Format	.mdc + frontmatter	SKILL.md + frontmatter	Düz markdown
Zorunluluk	Always Apply ile zorunlu	Opsiyonel	Bağlama göre otomatik
Kapsam	Proje/takım/kullanıcı	Proje	Proje/dizin
En iyi kullanım	Kısıtlar, standartlar	Tekrarlayan workflow'lar	Basit proje talimatları
Tab/Cmd+K etkisi	Hayır	Hayır	Hayır

24. Mermaid Mimari Diyagramları

24.1. Classic IDE Bileşen Etkileşimi

```
graph TB
    subgraph "Classic IDE (Editor Window)"
        EXPLORER["Explorer Panel"]
        EDITOR["Code Editor"]
        TERMINAL["Terminal Panel"]
        AGENT_PANEL["Agent Side Panel"]
        TAB_UI["Tab Ghost Text"]
    end

    subgraph "AI Pipeline"
        TAB_ENGINE["Tab Engine"]
        INLINE_ENGINE["Inline Edit Engine"]
        AGENT_ENGINE["Agent Orchestrator"]
    end

    subgraph "Context Layer"
        RULES_ENG["Rules Engine"]
        INDEX_ENG["Index/Search"]
        MCP_ENG["MCP Client"]
    end

    subgraph "Backend"
        MODELS["LLM Providers"]
        VECTOR_DB["Vector Store"]
    end

    EDITOR --> TAB_UI
    TAB_UI --> TAB_ENGINE
    EDITOR --> INLINE_ENGINE
    AGENT_PANEL --> AGENT_ENGINE

    TAB_ENGINE --> MODELS
    INLINE_ENGINE --> MODELS
```

```
AGENT_ENGINE --> RULES_ENG
AGENT_ENGINE --> INDEX_ENG
AGENT_ENGINE --> MCP_ENG
AGENT_ENGINE --> MODELS
INDEX_ENG --> VECTOR_DB
```

24.2. Tool Selection Decision Tree

```
graph TD
    START{"Ne yapmak<br/>istiyorsunuz?"}

    START -->|"Kod tamamlama"| TAB["Tab<br/>~320ms"]
    START -->|"Tek dosya düzenleme"| INLINE["Inline Edit<br/>Cmd+K"]
    START -->|"Çok dosyalı görev"| AGENT["Agent<br/>Cmd+I"]
    START -->|"Soru sorma"| CHAT["Chat Ask Mode<br/>Cmd+L"]
    START -->|"Hata ayıklama"| DEBUG["Debug Mode"]
    START -->|"Alternatif keşfi"| PARALLEL["Parallel Agents"]
    START -->|"Harici sistem"| MCP["MCP Tools"]

    style START fill:#e1f5ff,stroke:#0066cc,stroke-width:2px
    style TAB fill:#d4edda
    style INLINE fill:#fff4e1
    style AGENT fill:#fff3cd
    style CHAT fill:#cfe2ff
    style DEBUG fill:#ffe4e1
    style PARALLEL fill:#f0e1ff
    style MCP fill:#e1f5e1
```

24.3. İndeksleme ve Arama Akışı

```
sequenceDiagram
    participant U as Kullanıcı
    participant A as Agent
    participant I as Index Service
    participant V as Vector DB
    participant F as Yerel Dosya Sistemi
    participant L as LLM

    U->>A: "@codebase auth nasıl çalışıyor?"
    A->>I: Semantik arama isteği
    I->>I: Sorgu embedding hesapla
    I->>V: En yakın komşu araması
    V-->>I: Gizlenmiş path + satır aralıkları
    I->>F: Yerel dosyaları oku
    F-->>I: Kod içerikleri
    I->>L: Bağlam + soru
    L-->>A: Analiz yanıtı
    A-->>U: Auth akışı açıklaması
```

24.4. MCP Tool Call Akışı

```
sequenceDiagram
    participant A as Agent
    participant C as MCP Client
    participant S as MCP Server
    participant E as Harici Sistem
    participant U as Kullanıcı

    A->>C: Tool call önerisi (github.create_issue)
    C->>U: Onay iste
    U->>C: Onayla
    C->>S: MCP protocol call
    S->>E: GitHub API
    E-->>S: API yanıtı
    S-->>C: MCP response
    C-->>A: Tool sonucu
    A->>A: Sonucu değerlendir
```

24.5. Rules Birleştirme Önceliği

```
graph LR
    TR["Team Rules<br/>Öncelik 1"] --> MERGE["Birleştirilmiş<br/>Kural Seti"]
    PR["Project Rules<br/>Öncelik 2"] --> MERGE
    UR["User Rules<br/>Öncelik 3"] --> MERGE
    AG["AGENTS.md<br/>Alternatif"] --> MERGE
    MERGE --> PROMPT["System Prompt<br/>+ Kural İçeriği"]
    PROMPT --> LLM["LLM Inference"]
```

24. Derinlemesine Teknik Ekler

24.1. Electron ve VS Code Extension Host Mimarisi

Cursor Classic IDE, Electron framework üzerinde çalışan çok süreçli (multi-process) bir uygulamadır. VS Code mimarisinden devralınan süreç modeli şu bileşenleri içerir:

Ana Süreç (Main Process): Uygulama yaşam döngüsü, pencere yönetimi, native OS entegrasyonu ve güncelleme mekanizmasını yönetir. Classic IDE ve Agents Window pencereleri bu süreç tarafından oluşturulur.

Renderer Süreci: Her editör penceresi kendi renderer sürecinde çalışır. Classic IDE'nin Explorer, Editor, Terminal ve Agent paneli bu süreçte render edilir. AI özellikleri (Tab ghost text, Inline Edit diff overlay, Agent chat UI) renderer sürecinde görselleştirilir ancak hesaplama yoğunlukla ayrı servislere delege edilir.

Extension Host: VS Code eklentileri izole bir Node.js sürecinde çalışır. Language Server'lar (TypeScript, Python, Rust vb.), linter'lar, debugger adapter'ları ve tema eklentileri bu süreçte yürütülür. Cursor'ın AI katmanı, Extension Host'tan gelen LSP tanımlayıcıları ve diagnostic mesajlarını Tab ve Agent bağlamına besler.

Shared Process: Dosya izleme (file watcher), arama indeksi ve bazı arka plan görevleri paylaşımlı süreçte çalışır. Kod tabanı indeksleme pipeline'ının yerel dosya tarama aşaması bu sürece bağlıdır.

Bu süreç ayrımı, AI işlemlerinin editör yanıt verme süresini (responsiveness) bloke etmemesini sağlar. Tab önerileri arka planda hesaplanırken geliştirici yazmaya devam edebilir; Agent terminal komutu çalıştırırken editör kullanılabilir kalır.

24.2. Tab Modeli ve Fine-Tuning Detayları

Cursor Tab, genel amaçlı frontier modellerden farklı olarak kod tamamlama için optimize edilmiş özel bir model kullanır. Bu modelin eğitim verisi ve mimari tasarımı şu özelliklere odaklanır:

Edit Chain Prediction: Geleneksel autocomplete modelleri bir sonraki token'ı tahmin eder. Tab modeli bir sonraki *düzenleme eylemini* tahmin eder — bu, silme, ekleme, imleç hareketi ve hatta dosya değiştirme gibi eylemleri kapsayabilir. Bu yaklaşım "Copilot-style completion" ile "Cursor Tab" arasındaki temel farktır.

Recent Edit Context: Tab, oturum içindeki son düzenlemeleri (edit history) bağlam olarak kullanır. Bir fonksiyonu refactor ettiyseniz, Tab bir sonraki fonksiyonun benzer refactoring'ini önerebilir. Bu "pattern continuation" davranışı, Tab'ı tekrarlayan görevlerde son derece verimli kılar.

Linter-Aware Completion: Language Server'dan gelen diagnostic bilgisi Tab pipeline'ına beslenir. Eksik import, tip hatası veya undefined variable durumlarında Tab düzeltici öneriler sunabilir — salt metin tamamlama yerine anlamlı düzeltme önerileri.

Latency Budget: Tab'ın 320 ms gecikme hedefi, model boyutu (8B parametre), quantization, speculative decoding ve yerel/edge caching stratejileriyle sağlanır. Bu bütçe aşıldığında öneri gösterilmez; geliştirici deneyimi "yavaş öneri" yerine "öneri yok" tercih edilir.

24.3. Agent Tool Calling Protokol Uyumluluğu

Cursor Agent, farklı LLM sağlayıcılarının farklı tool calling formatlarını destekler:

Sağlayıcı	Tool Call Formatı	Cursor Adaptasyonu
Anthropic (Claude)	XML-tagged tool_use blocks	Native Anthropic API formatı
OpenAI (GPT)	function_call / tools API	OpenAI tools schema dönüşümü
Google (Gemini)	function calling API	Gemini-specific schema mapping
Cursor (Composer)	Özel format	Cursor-native optimizasyon

Cursor'ın orkestrasyon katmanı, modelden gelen tool call'ı standart bir iç forma dönüştürür, güvenlik kontrolünden geçirir, yürütür ve sonucu modelin beklediği formata geri çevirir. Bu abstraction sayesinde kullanıcı model değiştirdiğinde araç seti ve davranış tutarlı kalır.

24.4. Hooks: Yaşam Döngüsü Kontrolü

Cursor, `.cursor/hooks.json` ile Agent yaşam döngüsüne müdahale imkânı sunar. Hooks, Agent'ın ne zaman duracağını veya devam edeceğini belirleyen shell script'lerdir:

```
{
  "hooks": {
    "stop": {
      "script": "./scripts/stop-hook.sh",
      "description": "Tüm testler geçene kadar devam et"
    }
  }
}
```

```
}  
}
```

Stop hook exit code'ları: `0` = devam, `1` = dur. Bu mekanizma otonom TDD döngüleri, build verification loop'ları ve quality gate'ler için kullanılır. Hooks, Rules (kısıtlar) ve Skills (workflow'lar) ile birlikte üç katmanlı Agent yönlendirme sisteminin üçüncü bileşenidir.

24.5. Background Agent ve Classic IDE İlişkisi

Background Agent, bulut tabanlı bir Cursor hizmetidir; Classic IDE'den bağımsız olarak uzak bir ortamda görev yürütür. Classic IDE ile ilişkisi:

- Classic IDE'den Background Agent başlatılabilir
- Background Agent ilerlemesi Cursor dashboard ve IDE bildirimlerinden izlenir
- Sonuçlar (commit, PR) yerel repoya push edilir
- Team MCP sunucuları Background Agent tarafından da kullanılabilir

Background Agent, Classic IDE'nin yerel Agent'ından farklı olarak sandboxed bir bulut ortamında çalışır; geliştiricinin makinesi kapalıyken bile görev yürütebilir.

24.6. Customize Panel: Merkezi Yapılandırma

Cursor 2.0+ ile **Customize** paneli (sidebar), Rules, MCP, Skills, Commands ve eklenti yönetimini tek noktada birleştirir. Classic IDE'de Customize paneline sidebar'dan erişilir. Bu panel:

- Tüm MCP sunucularını listeler (proje + global + takım)
- Rules durumunu gösterir (aktif/pasif/enforced)
- Marketplace'ten tek tıkla MCP/eklenti kurulumu sağlar
- Team Rules'ın enforce durumunu görüntüler

Customize, Classic IDE'nin "Control Plane" arayüzü olarak düşünülebilir — AI davranışının yapılandırıldığı merkezi nokta.

24.7. Performans Profili ve Kaynak Kullanımı

Classic IDE'nin tipik kaynak profili:

Bileşen	CPU	Bellek	Disk I/O	Ağ
Tab Engine	Düşük-Orta	~200-500 MB	Minimal	Sürekli (düşük bant)
Indexing	Orta (ilk tarama)	~500 MB-2 GB	Yüksek (ilk)	Artımlı upload
Agent (aktif)	Orta-Yüksek	~1-4 GB	Orta	Yüksek (model API)
MCP Servers	Değişken	Değişken	Değişken	Değişken
Extension Host	Değişken	~200-1000 MB	Düşük	LSP bağlantıları

Büyük monorepo'larda ilk indeksleme belirgin CPU ve disk I/O tüketir; sonraki oturumlar artımlı güncelleme ile çok daha hafiftir. `.cursorindexingignore` ile indeksleme kapsamını daraltmak performansı önemli ölçüde iyileştirir.

24.8. Çoklu Workspace ve Proje Yapılandırması

Classic IDE, VS Code'un multi-root workspace özelliğini destekler. Her kök dizin bağımsız indekslenir ve Agent tüm köklerin bağlamına erişebilir. Ancak:

- Git worktree tabanlı Paralel Agents multi-root'ta devre dışıdır
- Cloud Agents multi-root workspace desteklemez
- Her workspace'in kendi `.cursor/mcp.json` ve `.cursor/rules/` dizini olabilir

Multi-root senaryosunda Rules birleştirme, her kök dizinin kendi `.cursor/rules/` setini bağımsız değerlendirir; AGENTS.md nested desteği her kök için ayrı çalışır.

25. Operasyonel En İyi Uygulamalar

25.1. Classic IDE Günlük Workflow

1. **Sabah:** `cursor . --classic` ile projeyi Classic IDE modunda açın
2. **Kod yazımı:** Tab ile boilerplate; Cmd+K ile yerel refactoring
3. **Özellik geliştirme:** Plan Mode ile plan → Agent ile uygulama
4. **Commit:** AI commit message + manuel review
5. **PR:** Agent attribution ile PR oluşturma; Cursor Blame ile review

25.2. Monorepo Stratejisi

- `.cursorindexingignore` ile generated/vendor dosyalarını indekslemeden hariç tutun
- Dizin bazlı nested AGENTS.md ve `.cursor/rules/` kullanın
- `@folder` referansları ile kapsamı sınırlayın
- Explore subagent ile geniş aramalar yapın

25.3. Kurumsal Deployment

- Team Rules ile organizasyon standartlarını enforce edin
- MCP Allowlist ile onaylı sunucuları sınırlayın
- Privacy Mode veya Enterprise veri politikalarını değerlendirin
- Admin Dashboard'dan commit attribution kontrolü
- `.cursorignore` ile secrets erişimini kısıtlayın

25.4. Model Seçim Stratejisi

Faz	Önerilen Model	Gerekçe
Planlama	Claude Sonnet / Opus	Derin analiz, karmaşık talimatlar
Uygulama	Composer 2.5	Hız, çok dosyalı düzenleme
Review	Gemini 3.x Pro	Geniş bağlam, tam codebase analizi
Hızlı fix	Tab / Inline Edit	Minimum gecikme

25.5. Bağlam Yönetimi

- Gereksiz @mention kullanmayın; her mention token tüketir
- Büyük dosyaları @file yerine ilgili fonksiyon seçerek Cmd+K kullanın
- MCP yanıtlarını filtrelemek için spesifik tool isteyin
- Explore subagent ile ana bağlamı koruyun
- Plan Mode'da planı workspace'e kaydedin (takım paylaşımı)

26. Sık Sorulan Sorular

Classic IDE ile Agents Window arasındaki teknik fark nedir?

Aynı Electron uygulaması, aynı AI pipeline'ları. Fark yalnızca UI layout ve varsayılan odak noktasıdır. Tüm AI özellikleri (Tab, Cmd+K, Agent, MCP, Rules) her iki pencerede de kullanılabilir.

Kaynak kodum Cursor sunucularında saklanıyor mu?

Hayır. Kaynak kod düz metin olarak saklanmaz. Yalnızca embedding vektörleri ve gizlenmiş metadata bulutta tutulur. Kod içeriği istemci tarafında okunur.

Rules Tab'ı etkiler mi?

Hayır. Rules yalnızca Agent (Chat) oturumlarında uygulanır. Tab ve Inline Edit (Cmd+K) Rules'tan bağımsız çalışır.

MCP sunucuları .cursorignore'daki dosyalara erişebilir mi?

Evet. `.cursorignore` AI dosya okuma araçlarını kısıtlar ancak terminal ve MCP araçları bu kısıtlamayı bypass edebilir. Hassas dosyalar için MCP erişimini de sınırlandırın.

VS Code eklentilerim Cursor'da çalışır mı?

Çoğu popüler eklenti Open VSX registry üzerinden çalışır. Microsoft'a özel eklentiler (Pylance, Remote-*) kısıtlı olabilir. Profil export veya manuel kopyalama ile geçiş yapılabilir.

Checkpoint ile git arasındaki fark nedir?

Checkpoint Agent oturumuna özel, yerel, geçici geri alma noktasıdır. Git kalıcı versiyon kontrolüdür. Checkpoint'ler takımla paylaşılmaz; git commit'leri paylaşılır.

Privacy Mode indekslemeyi tamamen durdurur mu?

Privacy Mode bulut indekslemesini devre dışı bırakır veya kısıtlar; embedding'ler yerelde tutulur. Tam offline çalışma model sağlayıcısına bağlı olarak değişir.

Agent terminal komutlarını onaysız çalıştırabilir mi?

Varsayılan olarak hayır. Run Modes ile güvenilir komutlar allowlist'e alınarak otomatik çalıştırılabilir. Enterprise ortamlarda bu sıkı tutulmalıdır.

27. Sonuç ve Mimari Çıkarımlar

Cursor Desktop Classic IDE, VS Code olgunluğu ile çok katmanlı AI orkestrasyonunu birleştirir.

Temel çıkarımlar:

- Orkestrasyon platformu** — LLM tek başına karar vermez; istemci araçları, sandbox ve bağlamı yönetir.
- Spektrum tasarımı** — Tab (RL, 320ms) → Fast Apply (1000 tok/s) → Agent (harness) → MCP (harici).
- Güvenli indeksleme** — Merkle + simhash + content proof; plaintext kod bulutta saklanmaz.
- Real-time RL** — Tab ve Composer üretimden öğrenir; farklı döngü hızları (~2s vs ~5s).
- Dynamic context** — Token verimliliği için dosya primitive'i; MCP %47 token tasarrufu.

6. **Classic = Agents backend** — UI farkı; api2.cursor.sh ve pipeline ortaktır.

2026+ yönler: Window Restoration netleştirmesi, native Windows sandbox, Instant Grep olgunluğu, Composer checkpoint sıklığı artışı, /goal otonom iterasyon.

Resmi dokümantasyon: <https://docs.cursor.com/>

28. Derinlemesine Teknik Ekler

Bu bölüm, operasyonel mühendislik ve platform ekipleri için ek derinlik sağlar.

28.1. IPC ve süreçler arası iletişim

Electron tabanlı editörlerde süreç izolasyonu güvenlik ve kararlılık için zorunludur. Classic IDE renderer sürecinde çalışan UI kodu, doğrudan dosya sistemine veya ağa erişemez; bu erişim Extension Host veya Main process üzerinden RPC ile gerçekleşir. Cursor, VS Code'un kanıtlanmış MessagePort mimarisini devralır ve AI katmanını bu hattın üzerine ekler. Tab tuş vuruşu renderer'da yakalanır, debounce edilir, ardından Cursor extension'a iletilir; extension bağlamı derleyip Connect-RPC isteği oluşturur. Yanıt stream'i geldikçe ghost text güncellenir. Bu pipeline'da her adımın gecikme bütçesi vardır: debounce ~50-100ms, protobuf encode ~1ms, RTT ~20-200ms (bölgeye göre), decode+render ~5ms. Toplam ~320ms hedefi aşılsa öneri gösterilmez — yavaş öneri, öneri yok tercih edilir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde

state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

IPC ve süreçler arası iletişim konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), IPC ve süreçler arası iletişim için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.2. Connect-RPC hata kodları ve yeniden deneme

Connect-RPC, gRPC status kodlarına benzer hata semantiği taşır. `unavailable` geçici ağ kesintisinde `exponential backoff` ile yeniden denenir. `resource_exhausted` rate limit aşımında kullanıcıya upgrade veya bekleme mesajı gösterilir. `unauthenticated` JWT süresi dolduğunda refresh token akışı tetiklenir; refresh başarısızsa kullanıcı yeniden login olmalıdır. Classic IDE bu akışları arka planda yönetir; Agents Window ile identiktir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine

bakılmalıdır. Bu alt başlık (3/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Connect-RPC hata kodları ve yeniden deneme konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Connect-RPC hata kodları ve yeniden deneme için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.3. Tab RL batch istatistikleri

Policy gradient güncellemesi gürültülü olduğundan Cursor büyük batch'ler toplar. RecordCppFate RPC her kabul/red olayını loglar. Batch yeterli büyüklüğe ulaşınca trainer checkpoint üretir. A/B test ile yeni checkpoint küçük yüzde trafiğe açılır; accept rate ve latency metrikleri izlenir. Regresyon varsa rollback anında yapılır. Bu operasyonel disiplin, real-time RL'nin production'da güvenli kalmasını sağlar.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback

prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Tab RL batch istatistikleri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Tab RL batch istatistikleri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.4. Speculative edits edge case'leri

Dosyanın tamamının değiştiği durumlarda speculative edits avantajı azalır — model her chunk'ta ayırır. Küçük cerrahi patch'lerde avantaj maksimumdur. Unicode, satır sonu normalizasyonu (CRLF vs LF) ve encoding (UTF-8 BOM) chunk sınırlarını etkileyebilir; Fast Apply eğitim verisi bu edge case'leri kapsayacak şekilde genişletilmiştir. Classic IDE diff renderer, satır sonu farklarını kullanıcıya gösterir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Speculative edits edge case'leri konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Speculative edits edge case'leri için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.5. Merkle senkronizasyon protokolü

İstemci ve sunucu Merkle kök hash'lerini karşılaştırır. Farklı dal bulunduğunda yalnızca o dalın yaprakları transfer edilir. Sunucu tarafında silinen dosyalar (istemcide yok) temizlenir; yeni dosyalar eklenir. Sync asla istemci dosyalarını değiştirmez — tek yönlü metadata/embedding güncellemesi. Privacy Mode'da sync kısıtlanabilir veya tamamen yerel moda geçilebilir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Merkle senkronizasyon protokolü konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Merkle senkronizasyon protokolü için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.6. Turbopuffer operasyonel parametreler

Namespace başına vektör boyutu embedding modeline bağlıdır (ör. 1536 veya 1024). ANN sorguları top-k parametresi ile Agent'a aday chunk listesi döndürür; k değeri harness'te model başına ayarlanır. Soğuk namespace warm-up gecikmesi büyük monorepo'larda ilk @codebase sorgusunu etkileyebilir; simhash reuse bu gecikmeyi dramatik azaltır.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek

bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . -` `-classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Turbopuffer operasyonel parametreler konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Turbopuffer operasyonel parametreler için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.7. Dynamic context dosya yolları

Tool spill dosyaları genellikle workspace `.cursor` alt dizininde veya OS temp altında oluşturulur. Chat history dosyası compaction öncesi yazılır; agent grep ile arama yapabilir. MCP tool sync klasöründe sunucu başına alt dizin yapısı, modelin tool keşfini insan dosya gezgini kullanımına benzetir. Terminal session dosyası PID veya session ID ile adlandırılır.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . - -classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Dynamic context dosya yolları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Dynamic context dosya yolları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.8. preCompact operasyonel rehber

Hook timeout süresi [GÖZLEM] ~120 saniye. Senkron işlem yapıyorsanız (transcript export, DB yazma) bu süreyi aşmamaya dikkat edin. Hook stdout'una JSON yazın: opsiyonel user_message kullanıcıya IDE bildirimi olarak gösterilir. Compaction iptal edilemediği için kritik state'i hook'a güvenmeyin — dosyaya persist edin.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

preCompact operasyonel rehber konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), preCompact operasyonel rehber için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.9. Seatbelt profil üretimi

Profil oluşturucu workspace root, `.cursorignore` glob'ları, admin deny list ve platform path kurallarını birleştirir. Symlink'ler [GÖZLEM] profil dışına taşmayı engellemek için resolve edilir. Ağ kuralları Cursor allowlist ile sınırlıdır: `api.github.com`, `api2.cursor.sh`, web search sağlayıcıları. Özel MCP HTTP sunucuları sandbox dışında veya explicit allow gerektirebilir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback

prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Seatbelt profil üretimi konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Seatbelt profil üretimi için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.10. Landlock overlay detayları

Overlay filesystem, ignored dosyaların sandbox görünümünde boş veya erişilemez stub ile değiştirilmesini sağlar. Bu, `.env` ve `secrets` için ikinci savunma hattıdır — `.cursorignore` tek başına yeterli değildir. Init süresi büyük workspace'lerde saniyeler sürebilir; agent ilk sandboxed komutta gecikme hissedilebilir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback

prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Landlock overlay detayları konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Landlock overlay detayları için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.11. state.vscdb bakım ve migrasyon

VS Code `state.vscdb` WAL modunda SQLite'dir. Cursor çalışırken veritabanı kilitli olabilir; harici araçlar `READ_ONLY + NO_MUTEX` bayrakları ile okunmalıdır. Profil export/import `state.vscdb`'yi taşımaz — token'lar makineye bağlıdır. BYOK secret'ları `safeStorage` ile OS keychain'e bağlıdır; farklı makineye taşınamaz.

`state.vscdb` bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), `state.vscdb` bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

`state.vscdb` bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), `state.vscdb` bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

`state.vscdb` bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), `state.vscdb` bakım ve migrasyon için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

state.vscdb bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), state.vscdb bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

state.vscdb bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), state.vscdb bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

state.vscdb bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), state.vscdb bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

state.vscdb bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), state.vscdb bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

state.vscdb bakım ve migrasyon konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), state.vscdb bakım ve migrasyon için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.12. Composer RL reward hacking

Real-time RL'de bilinen risk: model kısa vadeli reward'u maksimize edip uzun vadeli kaliteyi düşürmesi. Cursor CursorBench regresyon kapısı ve reward hacking incident'lerini public blog'da tartışmıştır. Classic IDE kullanıcıları Auto modda Composer checkpoint değişimlerini fark etmeyebilir — router arka planda günceller.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Composer RL reward hacking konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Composer RL reward hacking için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.13. Instant Grep indeks bakımı

Dosya değişikliğinde trigram indeks artımlı güncellenir; tam rebuild nadirdir. `.gitignore` ve `.cursorignore` uyumlu ignore kuralları indeks kapsamını daraltır. Binary dosyalar indekslenmez. Regex sorguları RE2 semantiğine yakındır; backtracking patlaması riski düşüktür.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback

prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore bypass` senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb schema` değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Instant Grep indeks bakımı konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Instant Grep indeks bakımı için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.14. Agent harness model başına farklar

Claude modelleri XML tool_use tercih eder; OpenAI JSON schema; Gemini function declaration. Cursor harness her model için ayrı system prompt, tool schema dönüşümü ve hata recovery mesajları içerir. Thinking modellerde reasoning block'ları bağlam tüketir; harness thinking çıktısını kullanıcıya gösterirken tool loop'tan ayırır. Classic IDE Agent paneli tüm modellerde aynı UI'ı sunar.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (`Cmd+Shift+P` → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Agent harness model başına farklar için SRE runbook perspektifinden ek

bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Agent harness model başına farklar konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Agent harness model başına farklar için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

28.15. Kurumsal deployment checklist

Team Rules enforce, MCP allowlist, Privacy Mode değerlendirmesi, commit attribution policy, `.cursorignore` secrets pattern, sandbox Run Mode sıklığı, Instant Grep/indeks kapsamı, BYOK key rotation prosedürü, audit hook (beforeShellExecution, afterFileEdit) — kurumsal Classic IDE rollout için minimum kontrol listesi.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (1/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (2/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde `state.vscdb` schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (3/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam

sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (4/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (5/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (6/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda `cursor . --classic` ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda `.cursorindexingignore` ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, `.cursorignore` bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (7/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Kurumsal deployment checklist konusunun operasyonel boyutu Classic IDE kullanıcı deneyimini doğrudan etkiler. Platform ekipleri bu davranışı izlerken Request ID (Cmd+Shift+P → Copy Request ID) toplamalı, MCP Logs ve Indexing durum panellerini kontrol etmelidir. Sorun reproduksiyonunda cursor . --classic ile minimal workspace açmak, Agents Window karışıklığını elimine eder. Gecikme profili Chrome DevTools Performance ve Cursor Process Explorer ile [GÖZLEM] ölçülebilir. Monorepo'larda .cursorindexingignore ve Hierarchical Cursor Ignore birlikte kullanılmalıdır. Güvenlik denetimlerinde sandbox profili, .cursorignore bypass senaryoları (MCP, terminal) ve BYOK key storage ayrı ayrı değerlendirilmelidir. Versiyon yükseltmelerinde state.vscdb schema değişiklikleri nadirdir ancak extension compatibility matrisine bakılmalıdır. Bu alt başlık (8/8), Kurumsal deployment checklist için SRE runbook perspektifinden ek bağlam sağlar: alerting eşikleri, rollback prosedürü, kullanıcı iletişim şablonu ve post-mortem checklist maddeleri kurumsal ortamlarda standardize edilmelidir.

Ek A: Terimler Sözlüğü (Genişletilmiş)

Terim	İngilizce	Açıklama
Classic IDE	Editor Window	VS Code düzenli editör penceresi
Agents Window	Agents Window	Sohbet-öncelikli pencere
Tab	Tab / Cpp	RL destekli otomatik tamamlama
Fast Apply	Fast Apply / Instant Apply	Speculative edits ile dosya uygulama
Harness	Agent harness	Model başına prompt+tool paketi
Connect-RPC	Connect-RPC	Buf protobuf-over-HTTP/2 protokolü
aiserver.v1	aiserver.v1	Cursor backend protobuf package
Merkle tree	Merkle tree	SHA-256 dosya/klasör hash ağacı
Simhash	Similarity hash	İndeks paylaşımı için özet fingerprint
Content proof	Content proof	Arama sonucu yetki kanıtı
Turbopuffer	Turbopuffer	Vektör veritabanı backend
Instant Grep	Instant Grep	Trigram indeksli regex arama
Shadow Workspace	Shadow workspace	Gizli LSP doğrulama penceresi
Seatbelt	sandbox-exec	macOS agent sandbox
Landlock	Landlock	Linux FS sandbox kernel API
preCompact	preCompact hook	Compaction öncesi gözlem hook'u
Context ring	Context ring	Dönüşümlü agent bağlam penceresi
BYOK	Bring Your Own Key	Kullanıcı API anahtarı
state.vscdb	state.vscdb	SQLite globalStorage veritabanı
Dynamic context	Dynamic context discovery	Dosya tabanlı bağlam keşfi

Speculative edits	Speculative edits	Dosya-draft tabanlı hızlandırma
Real-time RL	Real-time RL	Üretim token'larından RL

Ek B: Sürüm ve Uyumluluk Matrisi

Cursor	VS Code base	Classic bayrak	preCompact hook	Instant Grep	Agent sandbox
3.0.x	~1.9x	--classic	Hayır	Kısmi	Hayır
3.10+	~1.9x	--classic	Evet	Evet	Beta
3.11+	~1.9x	--classic	Evet	Evet	macOS/Linux
3.12+ [plan]	TBD	Window Restoration rename	Evet	Evet	+WSL2 Win

Özellik	Classic IDE	Agents Window	Paylaşılan?
Tab	Evet	Evet	Backend
Agent	Evet	Evet	Backend
İndeksleme	Evet	Evet	Aynı servis
Background Agent başlat	Evet	Evet	Aynı API
Layout	Editör merkez	Sohbet merkez	Hayır

Ek C: Uçtan Uca Veri Akış Diyagramları

C.1. Tab kabul/red → RL

```
sequenceDiagram
    participant U as Kullanıcı
    participant T as Tab Engine
    participant API as api2.cursor.sh
    participant RL as RL Pipeline

    T->>API: StreamCpp
    API-->>T: suggestion
    U->>T: Tab veya Escape
    T->>API: RecordCppFate (accept/reject)
    API->>RL: batch aggregate
    RL->>API: new checkpoint (hours)
```

C.2. @codebase semantik arama

```
flowchart TD
    A["@codebase sorgusu"] --> B["Query embedding"]
    B --> C["Turbopuffer ANN"]
    C --> D["Content proof filter"]
    D --> E["Obfuscated path + lines"]
    E --> F["İstemci yerel okuma"]
    F --> G["LLM bağlamı"]
```

C.3. Agent shell + sandbox

```
flowchart TD
    P["Model tool_call: shell"] --> APP{"Onay?"}
    APP -->|Hayır| STOP["Reddedildi"]
    APP -->|Evet| SB["Sandbox spawn"]
    SB --> MAC["macOS: sandbox-exec"]
    SB --> LIN["Linux: Landlock+seccomp"]
    MAC --> OUT["stdout → spill file"]
    LIN --> OUT
    OUT --> AG["Agent read/tail"]
```

Ek D: Protobuf ve SQL Referansları

D.1. RecordCppFate (Tab RL geri bildirimi) [TERS MÜHENDİSLİK]

```
message RecordCppFateRequest {
    string request_id = 1;
    CppFate fate = 2;
    optional int32 latency_ms = 3;
}
enum CppFate {
    CPP_FATE_UNSPECIFIED = 0;
    CPP_FATE_ACCEPT = 1;
    CPP_FATE_REJECT = 2;
    CPP_FATE_PARTIAL = 3;
    CPP_FATE_DISMISS = 4;
}
```

D.2. FSUpload (Merkle sync) [TERS MÜHENDİSLİK]

```
message FSUploadFileRequest {
    string relative_path = 1;
    string sha256_hash = 2;
    bytes encrypted_path = 3;
    optional bytes contents = 4; // yalnızca sync gerektiğinde
}
enum FSUploadErrorType {
```

```
FS_UPLOAD_ERROR_TYPE_HASH_MISMATCH = 2;
}
```

D.3. state.vscdb sorgu örnekleri

```
-- Access token okuma
SELECT value FROM ItemTable WHERE key = 'cursorAuth/accessToken';

-- Machine ID
SELECT value FROM ItemTable WHERE key = 'storage.serviceMachineId';

-- BYOK secret listesi
SELECT key FROM ItemTable WHERE key LIKE 'secret://%';

-- Chat metadata (boyut kontrolü)
SELECT length(value) FROM ItemTable
WHERE key LIKE 'workbench.panel.aichat%';
```

D.4. cursor_compactions şeması (hook telemetri örneği) [GÖZLEM]

```
CREATE TABLE cursor_compactions (
  id INTEGER PRIMARY KEY,
  conversation_id TEXT NOT NULL,
  trigger TEXT, -- auto | manual
  context_usage_percent REAL,
  context_tokens INTEGER,
  context_window_size INTEGER,
  message_count INTEGER,
  messages_to_compact INTEGER,
  is_first_compaction INTEGER,
  observed_at_ms INTEGER NOT NULL
);
```

Ek E: Doğrulama Durumu Matrisi

Konu	Durum	Kaynak
Tab reward $\pm 0.75/-0.25$	✓ Resmi	cursor.com/blog/tab-rl
Speculative edits ~1000 tok/s	✓ Resmi	cursor.com/blog/instant-apply
Merkle + simhash + content proof	✓ Resmi	cursor.com/blog/secure-codebase-indexing
Composer 5h RL cycle	✓ Resmi	cursor.com/blog/real-time-rl-for-composer

Dynamic context paths	✅ Resmi	cursor.com/blog/dynamic-context-discovery
Instant Grep trigram index	✅ Resmi	cursor.com/blog/fast-regex-search
Seatbelt / Landlock sandbox	✅ Resmi	cursor.com/blog/agent-sandboxing
Shadow Workspace gRPC	⚠️ Resmi (gRPC kullanımı); proto detayı [TERS MÜHENDİSLİK]	cursor.com/blog/shadow-workspace
aiserver.v1 RPC listesi	⚠️ [TERS MÜHENDİSLİK]	unite.proto , cursor_api_demo
state.vscdb key listesi	⚠️ [TERS MÜHENDİSLİK]	Topluluk auth araçları
IPC kanal adları	⚠️ [GÖZLEM]	VS Code + bundle analizi
Context ring iç yapısı	⚠️ [GÖZLEM]	Davranışsal
x-cursor-checksum algoritması	⚠️ [TERS MÜHENDİSLİK]	cursor_api_demo

Belge Sonu

Alan	Değer
Dosya (MD)	/workspace/docs/cursor-desktop-ultimate-tr.md
Dosya (PDF)	/opt/cursor/artifacts/cursor-desktop-ultimate-tr.pdf
Dil	Türkçe
Son güncelleme	Ağustos 2026