

# Claude Code Statusline Araçları Karşılaştırması

Claude Code statusline ekosisteminin kaynaklara dayalı bir karşılaştırması: framework'ler, plugin'ler, shell script'ler, Rust binary'leri, Ghostty'de hızlı bakış için en iyi düzenler ve pazarı hâlâ şekillendiren platform eksikleri.

comparison

developer-tools

Tarih: 2026-04-16

24 repo + 2 gist

[anthropics/claude-code](https://github.com/anthropics/claude-code) (<https://github.com/anthropics/claude-code>)

[ccstatusline](https://github.com/sirmalloc/ccstatusline) (<https://github.com/sirmalloc/ccstatusline>) [claude-powerline](https://github.com/Owloops/claude-powerline) (<https://github.com/Owloops/claude-powerline>) [cship](https://github.com/stephenleo/cship) (<https://github.com/stephenleo/cship>) [claudeline](https://github.com/fredrikaverpil/claudeline) (<https://github.com/fredrikaverpil/claudeline>) [claude-hud](https://github.com/jarrodwatts/claude-hud) (<https://github.com/jarrodwatts/claude-hud>)

## Yönetici Özeti

- Artık "tek bir en iyi Claude Code statusline" yok. En az beş ayrı ürün şekli var:
  - konfigürasyon framework'ü: [ccstatusline](https://github.com/sirmalloc/ccstatusline) (<https://github.com/sirmalloc/ccstatusline>)
  - plugin-first, theme odaklı powerline: [claude-powerline](https://github.com/Owloops/claude-powerline) (<https://github.com/Owloops/claude-powerline>)
  - Rust performans yaklaşımı: [CCometixLine](https://github.com/Haleclipse/CCometixLine) (<https://github.com/Haleclipse/CCometixLine>)

-

- Starship köprüsü: **cship** (<https://github.com/stephenleo/cship>)
- transcript-aware operasyon HUD'ı: **claude-hud** (<https://github.com/jarrodwatts/claude-hud>)
- **ccstatusline** (<https://github.com/sirmalloc/ccstatusline>) hâlâ kategori lideri çünkü çoğu kullanıcı için en okunabilir genel amaçlı cevap o. En güçlü widget yüzeyine, en bilinen TUI editörüne ve en erişilebilir default deneyime sahip.
- Ekosistemde ayrışma artık "model, context, git ve cost gösterebiliyor mu?" seviyesinde değil. Asıl farklar şu eksenlerde oluşuyor:
  - nasıl kurulduğu
  - hangi veriye güvendiği
  - render anında network ya da transcript işi yapıp yapmadığı
  - terminal estetiğine ne kadar yatırım yaptığı
- Resmi Claude Code desteği artık o kadar gelişti ki eski dönemin birçok hack'i mecburi olmaktan çıktı. `statusLine` payload'ı zaten model, context, cost, worktree, agent ve `rate_limits` verisini veriyor. Ama açık issue kuyruğu, platformun hâlâ ciddi şekilde istenen alanlar ve refresh garantileri eksik olduğunu gösteriyor.
- Günlük kullanımda test ettiğim en tatmin edici hızlı bakış düzeni hâlâ bir framework değil. **SippieCup'ın 30 Mart 2026 tarihli gist** (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>) çizgisine en yakın shell yaklaşımı, özellikle Ghostty için biraz daha iyi ayarlanmış haliyle, hâlâ en hoş görünen aile.

*"Kısa cevap: Hiç fazla düşünmeden tek öneri istiyorsan **ccstatusline** (<https://github.com/sirmalloc/ccstatusline>) kullan. Plugin-native ve daha temiz bir kurulum akışı istiyorsan **claude-powerline** (<https://github.com/Owloops/claude-powerline>) kullan. "Claude şu an ne yapıyor?" senin için görsel şıklıktan daha önemliyse **claude-hud** (<https://github.com/jarrodwatts/claude-hud>) kullan."*

## Dahil Etme Kriterleri ve Pazar Tanımı

Bu rapor, ařağıdaki kořullardan en az birini karřılayan herkese açık GitHub projelerini kapsıyor:

- doğrudan Claude Code statusline olarak üretilmiş olması
- Claude Code'un `statusLine` hook'una birinci sınıf özellik olarak kurulması
- pratikte kullanıcıların gerçekten statusline seçeneğı gibi değerlendirdiğı kadar yakın olması; örneğın **ccusage** (<https://github.com/ryoppippi/ccusage>)'ın beta `statusline` komutu ya da **claude-hud** (<https://github.com/jarrodwatts/claude-hud>) gibi HUD tarzı çözümler

Amaç, internetteki her kişisel `~/ .claude/statusline.sh` dosyasını toplamak değıl. Kamuya açık dokümantasyonu, install rehberi ya da gerçek seçenek olarak sayılmasını sağlayacak kadar görünürlüğü olan projelere odaklandım.

## Pazar Haritası

**ccstatusline** (<https://github.com/sirmalloc/ccstatusline>) **claude-powerline** (<https://github.com/Owloops/claude-powerline>) **CCometixLine** (<https://github.com/Haleclipse/CCometixLine>) **cship** (<https://github.com/stephenleo/cship>)

**claude-hud** (<https://github.com/jarrodwatts/claude-hud>) **claudeline** (<https://github.com/fredrikaverpil/claudeline>)

**kamranahmedse/claude-statusline** (<https://github.com/kamranahmedse/claude-statusline>)  
**daniel3303/ClaudeCodeStatusLine** (<https://github.com/daniel3303/ClaudeCodeStatusLine>)  
**felipeelias/claude-statusline** (<https://github.com/felipeelias/claude-statusline>) **claude-pace** (<https://github.com/Astro-Han/claude-pace>)  
**kcchien/claude-code-statusline** (<https://github.com/kcchien/claude-code-statusline>) **pyccsl** (<https://github.com/wolfdenpublishing/pyccsl>)  
**chongdashu/cc-statusline** (<https://github.com/chongdashu/cc-statusline>)

**pcvelz/ccstatusline-usage** (<https://github.com/pcvelz/ccstatusline-usage>) **syoun6162/ccstatusline** (<https://github.com/syoun6162/ccstatusline>)  
**FlineDev/CustomStatusline** (<https://github.com/FlineDev/CustomStatusline>) **ohugonnot/claude-code-statusline** (<https://github.com/ohugonnot/claude-code-statusline>) **xleddyl/claude-watch** (<https://github.com/xleddyl/claude-watch>)  
**rz1989s/claude-code-statusline** (<https://github.com/rz1989s/claude-code-statusline>)  
**Wangnov/claude-code-statusline-pro** (<https://github.com/Wangnov/claude-code-statusline-pro>)  
**sotayamashita/claude-code-statusline** (<https://github.com/sotayamashita/claude-code-statusline>)  
**daliovic/cc-statusline** (<https://github.com/daliovic/cc-statusline>) **leeguoooo/claude-code-**

usage-bar (<https://github.com/leeguoooo/claude-c>  
ode-usage-bar)

## İlk Dalgadan Bu Yana Ne Değişti

İlk public statusline örnekleri büyük ölçüde shell script'lerden oluşuyordu; çünkü platformda eksik veri vardı. Bugünkü pazar farklı çünkü resmi Claude Code payload'ı artık çok daha zengin:

- `model`
- `cost`
- `context_window`
- `agent`
- `worktree`
- `rate_limits`
- çok satırlı ANSI output ve OSC 8 link desteği

Bu, problemin şeklini değiştirdi. 2026'nın ilginç statusline araçları artık sadece "stdin'i güzel parse ediyor mu?" diye ayrışmıyor. Asıl ayrışma konfigürasyon UX'i, ek veri kaynakları, güvenlik duruşu ve görsel hiyerarşi tarafında.

## Resmi Claude Code Statusline Platformunun Temeli

Resmi **Claude Code statusline docs** (<https://docs.anthropic.com/en/docs/claude-code/statusline>) çok basit bir kontrat tanımlıyor:

- Claude Code, `settings.json` içinden bir shell komutu çalıştırıyor
- komuta stdin üzerinden JSON gönderiyor

-

- senin komutun string ya da çok satırlı bir blok basıyor
- ANSI renkler ve OSC 8 linkler destekleniyor

Ekosistemin bu kadar hızlı büyümesinin sebebi tam olarak bu basitlik.

## Temel Kabiliyet Seti

16 Nisan 2026 itibarıyla resmi payload, çoğu ana akım statusline için zaten yeterince zengin:

- `model.id` ve `model.display_name` ile model kimliği
- `cost.*` ile session cost ve code-change istatistikleri
- `context_window.*` ile context size ve current usage
- `rate_limits.five_hour` ve `rate_limits.seven_day` ile rate-limit pencereleri
- workspace ve worktree bilgisi
- agent adı
- output style ve vim mode

Ürün seviyesinde iki önemli kolaylık da var:

- `/statusline`, doğal dilden statusline scaffold edebiliyor
- resmi docs çok satırlı output'u destekliyor; bu yüzden artık iki ya da üç satırlı tasarımlar hack değil, doğal desen

## Bunun Pratik Sonucu

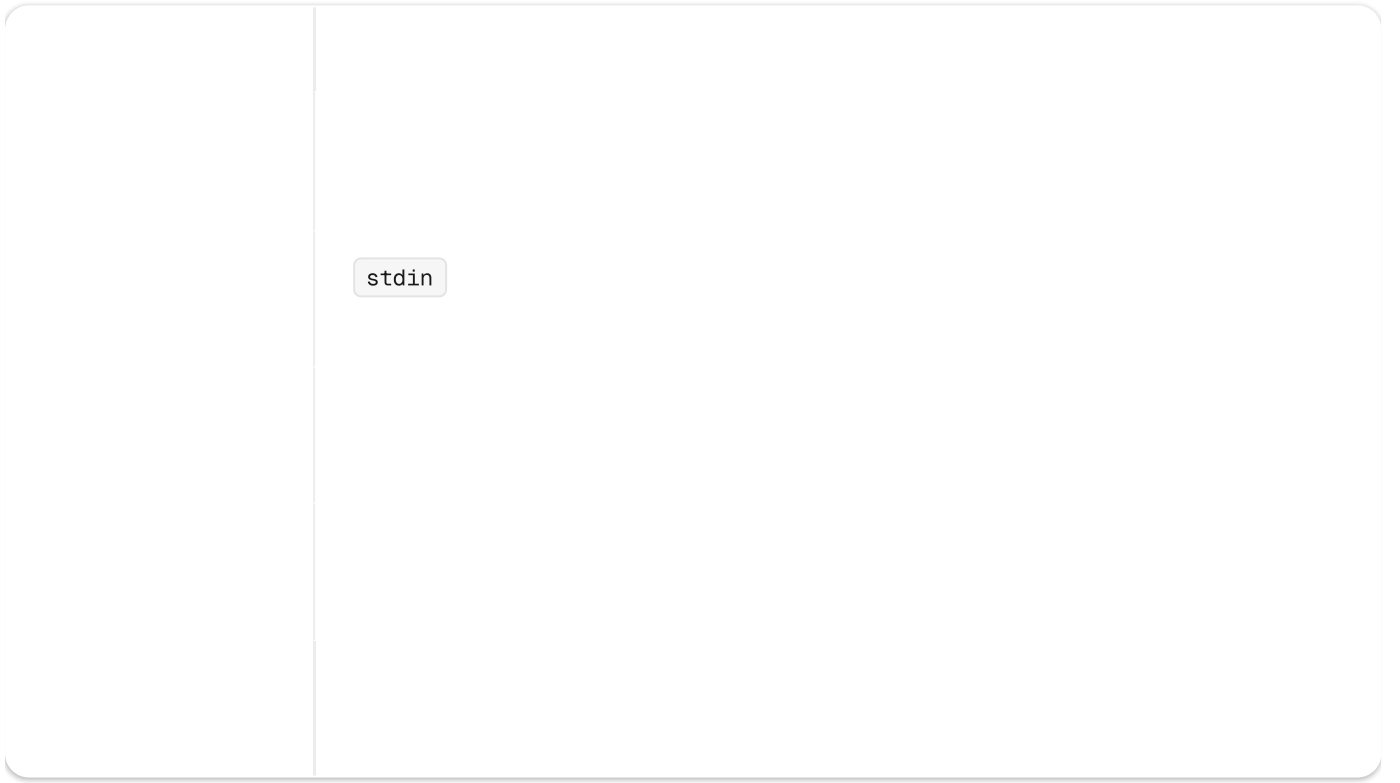
Sadece model, context, cost ve subscriber limitlerini göstermek istiyorsan artık credential scrape etmene ya da log parse etmene gerek yok. Ek iş yapan araçlar bunu genelde şu dört sebepten biri için yapıyor:

- daha zengin ya da daha tutarlı usage verisi almak
-

- canlı tool ya da subagent aktivitesi göstermek
- cache efficiency, burn rate gibi daha derin session analitiđi üretmek
- daha iyi install ve config UX'i sunmak

## Karşılaştırma Çerçevesi ve Sinyal Efsanesi

Bu raporda projeleri beş lens üzerinden kıyaslıyorum.



Aşağıda kullandığım sinyal etiketleri:

- **Leader** : yüksek benimsenme + gerçekten farklılaşan ürün şekli
- **Specialist** : belirli kullanıcı ya da workflow için çok net değer
- **Appendix** : ilginç ama daha dar, daha yeni, daha küçük ya da ana katmanla fazla örtüşen proje

# 1. Katman: Framework'ler, Plugin'ler ve Tam Sistemler

## Ana Karşılaştırma

<p>sirmalloc/ccstatusline (<a href="https://github.com/sirmalloc/ccstatusline">https://github.com/sirmalloc/ccstatusline</a>)</p>			<code>npx</code> <code>bunx</code>	
<p>Owloops/claude-powerline (<a href="https://github.com/Owloops/claude-powerline">https://github.com/Owloops/claude-powerline</a>)</p>			<code>npx</code>	
<p>Haleclipse/CCometixLine (<a href="https://github.com/Haleclipse/CCometixLine">https://github.com/Haleclipse/CCometixLine</a>)</p>				
<p>stephenleo/cship (<a href="https://github.com/stephenleo/cship">https://github.com/stephenleo/cship</a>)</p>			<code>cargo</code> <code>install</code>	
<p>jarrodwatts/claude-hud (<a href="https://github.com/jarrodwatts/claude-hud">https://github.com/jarrodwatts/claude-hud</a>)</p>				
<p>fredrikaverpil/claodeline (<a href="https://github.com/fredrikaverpil/claodeline">https://github.com/fredrikaverpil/claodeline</a>)</p>			<code>go install</code>	

## Kabiliyet Matrisi

-

CCSTATUSLINE (HTTPS://GITHUB.COM/SIRMALLOC/CCSTATUSLINE)

CLAUDE-POWERLINE

## [Her Birinci Katman Lideri Aslında Neyi Kazanıyor](#)

### [ccstatusline \(https://github.com/sirmalloc/ccstatusline\)](https://github.com/sirmalloc/ccstatusline)

[ccstatusline \(https://github.com/sirmalloc/ccstatusline\)](https://github.com/sirmalloc/ccstatusline) hâlâ kategori lideri çünkü bir script paketi gibi değil, gerçek bir ürün gibi davranıyor. En geniş widget yüzeyi, en bilinen TUI editörü, multi-line esnekliği, powerline desteği ve kendi config'ini elle yazmak istemeyen kullanıcılar için gerçekten erişilebilir bir default deneyimi var.

En büyük zayıflığı kabiliyet tarafında değil; operasyonel duruş tarafında. En kolay belgelenmiş yol hâlâ hareketli versiyonlu `npx` ya da `bunx` desenlerine yaslanıyor ve proje yüzeyi bazı kullanıcılar için ihtiyaçtan büyük gelebiliyor.

-

## [claude-powerline](https://github.com/Owloops/claude-powerline) (https://github.com/Owloops/claude-powerline)

Eğer [ccstatusline](https://github.com/sirmalloc/ccstatusline) (https://github.com/sirmalloc/ccstatusline) framework cevabıysa, [claude-powerline](https://github.com/Owloops/claude-powerline) (https://github.com/Owloops/claude-powerline) da plugin-native cevap. Bu araç önce Claude plugin'i, sonra statusline paketi gibi hissettiriyor: slash-command setup, config auto-reload, tema aileleri, Unicode ve ASCII modları ve utilitarian çizgiden daha bilinçli bir görsel zevk.

Güzel bir şey isteyen ama TUI içinde yaşamak istemeyen kullanıcı için en temiz önerilerden biri bu.

## [CCometixLine](https://github.com/Haleclipse/CCometixLine) (https://github.com/Haleclipse/CCometixLine)

[CCometixLine](https://github.com/Haleclipse/CCometixLine) (https://github.com/Haleclipse/CCometixLine), "Rust ile yazılmış bir [ccstatusline](https://github.com/sirmalloc/ccstatusline) (https://github.com/sirmalloc/ccstatusline)" değil. Daha farklı bir karakteri var. Rust statusline binary'si, built-in theme'ler, TUI, transcript tabanlı usage mantığı ve context warning kapatma ya da verbose mode patch etme gibi Claude Code yardımcıları aynı yerde.

Bu onu güçlü yapıyor, ama daha da geniş bir yüzeye taşıyor. Sadece formatter isteyenler render hızını sevebilir ama yanındaki yardımcı araç katmanını fazla bulabilir.

## [cship](https://github.com/stephenleo/cship) (https://github.com/stephenleo/cship)

[cship](https://github.com/stephenleo/cship) (https://github.com/stephenleo/cship), pazarın en temiz teknik tezine sahip: yeni bir config dili icat etmek yerine Starship mental modelini Claude Code içine taşımak. Halihazırda `starship.toml` yatırımın varsa, o yatırımın Claude Code içinde de devam etmesini sağlayan tek ciddi seçenek bu.

Bunun tradeoff'u da açık. Starship umurunda değilse, [cship](https://github.com/stephenleo/cship) (https://github.com/stephenleo/cship)'in dehasının bir kısmı senin için boşa gidiyor.

## [claude-hud](https://github.com/jarrodwatts/claude-hud) (https://github.com/jarrodwatts/claude-hud)

[claude-hud](https://github.com/jarrodwatts/claude-hud) (https://github.com/jarrodwatts/claude-hud) artık klasik anlamda statusline olmaktan biraz çıktı. Daha çok kompakt bir HUD. Önemli olmasının sebebi basit: resmi stdin payload'ı sana Claude'un tam şu anda hangi tool'u kullandığını, hangi subagent'ların

aktif olduğunu ya da todo listesinin nasıl ilerlediğini söylemiyor. [claude-hud](https://github.com/jarrodwatts/claude-hud) (<https://github.com/jarrodwatts/claude-hud>) bunu transcript JSONL okuyarak getiriyor.

"Claude şu an ne yapıyor?" sorusu senin için görsel sadelikten daha önemliyse, bu alanın en net ürün değerine sahip çözümü [claude-hud](https://github.com/jarrodwatts/claude-hud) (<https://github.com/jarrodwatts/claude-hud>).

## [claudeline](https://github.com/fredrikaverpil/claudeline) (<https://github.com/fredrikaverpil/claudeline>)

[claudeline](https://github.com/fredrikaverpil/claudeline) (<https://github.com/fredrikaverpil/claudeline>), star sayısının düşündürdüğünden daha güçlü bir mimari hikâyeye sahip küçük bir proje. Go stdlib binary, offline capture/render workflow, subscription plan ile provider ayrımı, service disruption göstergesi ve cache TTL tarafında bilinçli kararlar var.

Operasyonel açıdan sade ve anlaşılır bir çözüm isteyen altyapı kafalı kullanıcıya önereceğim proje bu olurdu.

## [Birinci Katman Özeti](#)

- En iyi genel default: [ccstatusline](https://github.com/sirmalloc/ccstatusline) (<https://github.com/sirmalloc/ccstatusline>)
- En iyi plugin-first setup: [claude-powerline](https://github.com/Owloops/claude-powerline) (<https://github.com/Owloops/claude-powerline>)
- En iyi Rust ürün yüzeyi: [CCometixLine](https://github.com/Haleclipse/CCometixLine) (<https://github.com/Haleclipse/CCometixLine>)
- Halihazırda Starship kullananlar için en iyisi: [cship](https://github.com/stephenleo/cship) (<https://github.com/stephenleo/cship>)
- En iyi gözlem katmanı: [claude-hud](https://github.com/jarrodwatts/claude-hud) (<https://github.com/jarrodwatts/claude-hud>)
- Operasyonel olarak en temiz minimal binary: [claudeline](https://github.com/fredrikaverpil/claudeline) (<https://github.com/fredrikaverpil/claudeline>)

## [2. Katman: Opinionated Orta Katman Araçlar](#)

Bu projeler pazarın tamamını tanımlamıyor olabilir, ama zevk ve fikir tarafındaki en ilginç farklar genelde burada çıkıyor.

**kamranahmedse/claude-statusline**  
(<https://github.com/kamranahmedse/claude-statusline>)

~/ .claude/

**daniel3303/ClaudeCodeStatusLine**  
(<https://github.com/daniel3303/ClaudeCodeStatusLine>)

**felipeelias/claude-statusline** (<https://github.com/felipeelias/claude-statusline>)

**Astro-Han/claude-pace** (<https://github.com/Astro-Han/claude-pace>)

**kcchien/claude-code-statusline** (<https://github.com/kcchien/claude-code-statusline>)

**wolfdenspublishing/pyccsl** (<https://github.com/wolfdenspublishing/pyccsl>)

chongdashedu/cc-statusline (<https://github.com/chongdashedu/cc-statusline>)

leeguooooo/claude-code-usage-bar (<https://github.com/leeguooooo/claude-code-usage-bar>)

## Özellikle Önemli Olan Orta Katmanlar

[kamranahmedse/claude-statusline](https://github.com/kamranahmedse/claude-statusline) (<https://github.com/kamranahmedse/claude-statusline>)

Bu, "güvendiğim birinin mantıklı default'unu istiyorum" seçeneği. Küçük, opinionated ve sıkıcı ama önemli işleri doğru yapıyor: script'i kopyalamak, settings'i patch etmek, eski durumu yedeklemek ve uninstall sunmak.

Bu, düşündüğünden daha değerli bir ürün şekli.

[daniel3303/ClaudeCodeStatusLine](https://github.com/daniel3303/ClaudeCodeStatusLine) (<https://github.com/daniel3303/ClaudeCodeStatusLine>)

Daniel Oliveira'nın repo'su, public setteki en iyi çapraz platform referans implementasyonlardan biri. Bash ve PowerShell'in beraber gelmesi önemli; çünkü birçok iyi statusline deposu Windows devreye girdiği an ciddiyetini kaybediyor.

Ayrıca "bu script'i Claude'a verip kendine kurdur" akışını da çok okunur hâle getiriyor.

[felipeelias/claude-statusline](https://github.com/felipeelias/claude-statusline) (<https://github.com/felipeelias/claude-statusline>)

Felipe'nin projesi benimsenme açısından küçük ama fikir olarak temiz: Go, TOML, preset'ler, preview komutları ve hyperlink desteği. Var olan araçların hiçbiri kendi zevkine uymadığı için Go ile prompt engine yazmış birinin işi gibi duruyor.

Henüz pazar lideri demem, ama küçük projeler içinde en derli toplu olanlardan biri.

### [claude-pace \(https://github.com/Astro-Han/claude-pace\)](https://github.com/Astro-Han/claude-pace)

[claude-pace \(https://github.com/Astro-Han/claude-pace\)](https://github.com/Astro-Han/claude-pace), star sayısından daha fazla ilgiyi hak ediyor çünkü problemi doğru çerçeveliyor. "Yüzde 60 kullandın" tek başına zayıf bilgi. Kalan süreye göre sürdürülebilir hızda gidip gitmediğini gösteren pace delta ise çok daha operasyonel bir ölçü.

Özellikle quota davranışıyla ilgileniyorsan ve saf Bash + `jq` istiyorsan, bu kategorinin en zeki dar araçlarından biri.

### [kcchien/claude-code-statusline \(https://github.com/kcchien/claude-code-statusline\)](https://github.com/kcchien/claude-code-statusline)

Bu repo büyüklüğünden bağımsız olarak tasarım referansı olarak önemli. Gradient context bar, truecolor fallback hikâyesi, smart hiding ve kompakt yoğunluk, "bir metrik daha sıkıştırılabilir" mantığından ziyade çevresel bakış okunabilirliğini önemseyen statusline tasarımına iyi örnek.

Kamuya açık set içinde "yoğun ama zevkli shell UI" örneklerinden biri.

### [pyccsl \(https://github.com/wolfdanpublishing/pyccsl\)](https://github.com/wolfdanpublishing/pyccsl)

[pyccsl \(https://github.com/wolfdanpublishing/pyccsl\)](https://github.com/wolfdanpublishing/pyccsl), sıfır dependency ile zengin metrik sunan Python statusline isteyen biri için en güçlü cevap. Cache hit rate, token breakdown, response time ve tema çeşitliliği, onu framework olmayan ama çok veri veren araçlar arasında üst sıraya koyuyor.

Ama hâlâ tek dosyalık bir Python dünyası. Bazı kullanıcı için artı, bazıları için otomatik eksi.

## [3. Katman: Shell Script'ler, Uzman Araçlar ve Uzun Kuyruk](#)

Uzun kuyruk önemli çünkü bu pazarda yenilik hâlâ framework'lerden çok shell script tarafında çıkıyor.

## Bilmeye Değer Uzun Kuyruk Projeleri

- [xleddyl/claude-watch](https://github.com/xleddyl/claude-watch) (https://github.com/xleddyl/claude-watch)
  - Büyük olduğu için değil, usage cache'i hook'larla arka planda yenileyip render'ı hızlı tutma desenini normalize ettiği için önemli.
- [FlineDev/CustomStatusline](https://github.com/FlineDev/CustomStatusline) (https://github.com/FlineDev/CustomStatusline)
  - Rate-limit izleme odaklı pluginize edilmiş bir yaklaşım. Tüm framework yerine usage pencerelerine odaklananlar için iyi referans.
- [ohugonnot/claude-code-statusline](https://github.com/ohugonnot/claude-code-statusline) (https://github.com/ohugonnot/claude-code-statusline)
  - `/oauth/usage` takibi ve reset countdown'ları için doğrudan, script-first bir cevap.
- [syou6162/ccstatusline](https://github.com/syou6162/ccstatusline) (https://github.com/syou6162/ccstatusline)
  - İsim benzerliğine rağmen farklı bir Go projesi. Sabit widget'lar yerine YAML ile tanımlanan shell aksiyonları ve cache yaklaşımı var.
- [sotayamashita/claude-code-statusline](https://github.com/sotayamashita/claude-code-statusline) (https://github.com/sotayamashita/claude-code-statusline)
  - Starship fikirlerini Claude'a özel gömülebilir Rust binary'sine taşıyan küçük bir proje.
- [Wangnov/claude-code-statusline-pro](https://github.com/Wangnov/claude-code-statusline-pro) (https://github.com/Wangnov/claude-code-statusline-pro)
  - Çok dilli, preset ağırlıklı ve basit shell script'ten daha iddialı. Ama bugünkü hâliyle kendi kitlesi dışında kategori belirleyici değil.
- [rz1989s/claude-code-statusline](https://github.com/rz1989s/claude-code-statusline) (https://github.com/rz1989s/claude-code-statusline)
  - Aşırı özellikli shell paketi. İlginç, ama feature kalabalığı her zaman kategori liderliği etmiyor.

-

- [daliovic/cc-statusline](https://github.com/daliovic/cc-statusline) (<https://github.com/daliovic/cc-statusline>)
  - Prayer times ve MCP tarzı yan sinyallerle farklılaşmaya çalışıyor.

## Uzun Kuyruk Neden Hâlâ Değerli

Uzun kuyruk, framework'lerin daha sonra kopyaladığı desenleri erken keşfediyor:

- hook tabanlı background refresh
- daha iyi SSH ya da fallback glyph hikâyeleri
- daha dürüst rate-limit semantiği
- daha akıllı context bar'lar
- yeni install desenleri

Star'ları framework'ler topluyor olabilir. Ama orijinal fikirlerin önemli kısmı hâlâ script dünyasından geliyor.

## Mimari Desenler ve Veri Kaynağı Soyları

Bu pazarı anlamamanın en faydalı yolu bence burası.

**claude-powerline** (<https://github.com/Owloops/claude-powerline>)  
**felipeelias/claude-statusline** (<https://github.com/felipeelias/claude-statusline>)  
**claude-pace** (<https://github.com/Astro-Han/claude-pace>)

**claudeline** (<https://github.com/fredrikaverpil/claudeline>)  
**daniel3303/ClaudeCodeStatusLine** (<https://github.com/daniel3303/ClaudeCodeStatusLine>)

**claude-hud** (<https://github.com/jarrodwatts/claude-hud>)  
**CCometixLine** (<https://github.com/Haleclipse/CCometixLine>)  
**pyccsl** (<https://github.com/wolfdenspublishing/pyccsl>)

**xleddyl/claude-watch** (<https://github.com/xleddyl/claude-watch>)

**chongdashu/cc-statusline** (<https://github.com/chongdashu/cc-statusline>)  
**kamranahmedse/claude-statusline** (<https://github.com/kamranahmedse/claude-statusline>)

**claude-powerline** (<https://github.com/Owloops/claude-powerline>)  
**claude-hud** (<https://github.com/jarrodwatts/claude-hud>)  
**claudeline** (<https://github.com/fredrikaverpil/claudeline>)  
**claude-pace** (<https://github.com/Astro-Han/claude-pace>)

`/api/oauth/`

`usage`

## En Büyük Soy Ayrımı

Tarihsel olarak en önemli ayırım şu:

- `rate_limits` öncesi statusline'lar çoğunlukla OAuth credential okuyup Anthropic API'sine direkt gidiyordu
- `rate_limits` sonrası statusline'lar resmi stdin verisini daha çok tercih ediyor ve ekstra işi sadece payload'ın vermediği özellikler için yapıyor

Bu yüzden modern statusline'lar artık hack değil, ürün tercihi gibi görünüyor. Baseline yeterince stabil; ek karmaşıklık çoğu zaman mecburiyetten değil, bilinçli iddiadan geliyor.

## Hızlı Bakış İçin Shell Soy: Ghostty Vaka Çalışması

Benim günlük kullanımda denediğim ondan fazla public statusline arasında en görsel olarak tatmin edici olanı hâlâ bir framework değil, ayarlanmış bir shell script oldu.

## Bu Düzen Neden Bu Kadar İyi Çalışıyor

- yakından okumaya değil, çevresel bakışa göre tasarlanmış
- current ve weekly aynı satırda olduğu için ilişkileri anında görülüyor
- renk geçişleri Ghostty'de dikkat çekecek kadar belirgin
- dev üçüncü satır context'i sayı olmaktan çıkarıp şekle dönüştürüyor
- bilgi yoğun olmasına rağmen çizgi sakın hissettiriyor

Bu kombinasyon sanıldığından daha nadir.

*“Saha notu: Şu ana kadar 10'dan fazla public Claude Code statusline varyasyonu denedim. Özellikle Ghostty'de hızlı bakış ve görsel tatmin açısından hâlâ en çok hoşuma giden çözüm bu çizgi oldu.”*

## Bulabildiğim En Yakın Public Upstream

Local script soyunun bulabildiğim en yakın public upstream'i **SippieCup**'ın 30 Mart 2026 tarihli **gist**'i (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>).

Kod benzerliği açısından en güçlü eşleşme buydu. Aynı ailede daha eski public akrabalar da var; özellikle **jtbr**'nin 8 Şubat 2026 tarihli **gist**'i (<https://gist.github.com/jtbr/4f99671d1cee06b44106456958caba8b>). Ama local script ile en güçlü örtüşme **SippieCup** (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>) tarafında çıktı.

Buradaki önemli nüans şu:

- bunun kusursuz bir tek-adımlı provenance zinciri olduğunu iddia etmiyorum
- sadece public olarak doğrulayabildiğim en yakın upstream'in **SippieCup** (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>) olduğunu söylüyorum

## Local Ghostty-Tuned Varyantta Neler Değişmiş

```
--dangerously-
```

```
skip-
```

```
permissions
```

```
□
```

Benim gördüğüm kadarıyla bu, "bir metrik daha sığdıralım" yerine "tek bakışta anlaşılabilir olsun" diye optimize edilmiş en iyi örneklerden biri.

# Güvenlik, Güvenilirlik ve Supply-Chain Riskleri

Ekosistemdeki en büyük açık eksik özellik değil. Update güven modeli.

## Temel Risk

Popüler projelerin bir kısmı hâlâ şu tip kurulum desenleri öneriyor:

- `npx -y ccstatusline@latest`
- `npx -y @owloops/claude-powerline@latest --style=powerline`
- `npx claude-pace`

Bunu resmi modelle birleştirdiğinde, yani Claude Code'un statusline komutunu tekrar tekrar çalıştırdığı gerçeğiyle, kaçınılmaz bir operasyon sorusu çıkıyor:

- sürekli çalışan bir UI hook'unun çalışma anında hareketli paket versiyonu çözmesini gerçekten istiyor musun?

Bu risk cümlesi, resmi hook kontratı ve README'lerdeki install örneklerinden çıkan bir çıkarım. Bilinen bir güvenlik olayı iddiası değil. Ama operasyonel tradeoff yine de gerçek.

## Güvenilirlik Matrisi

npx

**claudeline** (<https://github.com/fredrika-verpil/claudeline>) **CCometixLine** (<https://github.com/Haleclipse/CCometixLine>) **cship** (<https://github.com/stephenleo/cship>) **kamranahmedse/claude-statusline** (<https://github.com/kamranahmedse/claude-statusline>) **daniel3303/ClaudeCodeStatusLine** (<https://github.com/daniel3303/ClaudeCodeStatusLine>)

**ccstatusline** (<https://github.com/sirmaloc/ccstatusline>) **claude-powerline** (<https://github.com/Owloops/claude-powerline>) **claude-pace** (<https://github.com/Astro-Han/claude-pace>) npx

**claudeline** (<https://github.com/fredrika-verpil/claudeline>) **daniel3303/ClaudeCodeStatusLine** (<https://github.com/daniel3303/ClaudeCodeStatusLine>) **ohugonnot/claude-code-statusline** (<https://github.com/ohugonnot/claude-code-statusline>)

**claude-hud** (<https://github.com/jarrodwatts/claude-hud>) **CCometixLine** (<https://github.com/Haleclipse/CCometixLine>) **pyccsl** (<https://github.com/wolfdempublishing/pyccsl>)

**xleddyl/claude-watch** (<https://github.com/xleddyl/claude-watch>)

- En az drama istiyorsan, hot path içinde `npx @latest` yerine local binary ya da kopyalanmış script tercih et.
- Subscriber-limit verisini daha zengin istiyorsan cached API polling hâlâ mantıklı olabilir; ama sadece credential lookup ve cache TTL tarafı düzgün düşünölmüşse.
- Tool ve subagent görünürlüğü istiyorsan transcript parsing hâlâ tek ciddi yol; bunun da doğası gereği Claude Code internallerine daha yakın ve daha kırılğan olduğunu kabul etmek gerekiyor.

## Persona ve Kullanım Senaryosuna Göre Öneriler

**ccstatusline** (<https://github.com/sirmalloc/ccstatusline>)

**claude-powerline** (<https://github.com/Owloops/claude-powerline>)

**cship** (<https://github.com/stephenleo/cship>)

**CCometixLine** (<https://github.com/Haleclipse/CCometixLine>)

**claude-hud** (<https://github.com/jarrodwatts/claude-hud>)

**claudeline** (<https://github.com/fredrikaverpil/claudeline>)

**claude-pace** (<https://github.com/Astro-Han/claude-pace>)

**daniel3303/ClaudeCodeStatusLine** (<https://github.com/daniel3303/ClaudeCodeStatusLine>)

**pyccsl** (<https://github.com/wolfdenpublishing/pyccsl>)

kcchien/claude-code-statusline (<https://github.com/kcchien/claude-code-statusline>)

SippieCup (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>)

## [Açık Platform Boşlukları ve GitHub Issue Takip Listesi](#)

Açık [anthropics/claude-code](https://github.com/anthropics/claude-code) (<https://github.com/anthropics/claude-code>) kuyruğu, pazarın en net baskı noktalarını gösteriyor. 16 Nisan 2026 itibarıyla en çok izlenmeye değer issue'lar bunlar.

#48445 (<https://github.com/anthropics/claude-code/issues/48445>) `statusLine.refreshInterval`

#47071 (<https://github.com/anthropics/claude-code/issues/47071>)

#49022 (<https://github.com/anthropics/claude-code/issues/49022>) `context_breakdown`

#49270 (<https://github.com/anthropics/claude-code/issues/49270>)

#47534 (<https://github.com/anthropics/claude-code/issues/47534>)

#44982 (<https://github.com/anthropics/claude-code/issues/44982>)

#40279 (<https://github.com/anthropics/claude-code/issues/40279>)

#40287 (<https://github.com/anthropics/claude-code/issues/40287>) `/rename`

#37216 (<https://github.com/anthropics/claude-code/issues/37216>) `tmux`

Genel desen çok açık:

- builder'lar daha iyi repaint semantiği istiyor
- birkaç operasyonel alan daha görmek istiyor

-

- Unicode ve hyperlink davranışının terminaller arası daha öngörülebilir olmasını bekliyor

## Metodoloji ve Tam Kaynak Eki

### Metodoloji

- Araştırma tarihi: 16 Nisan 2026
- Birincil kaynak sırası:
  - resmi Claude Code docs
  - `gh` ile alınmış güncel GitHub repo metadata'sı
  - güncel proje README'leri
  - açık [anthropics/claude-code](https://github.com/anthropics/claude-code) (<https://github.com/anthropics/claude-code>) issue'ları
  - public gist'lerle yapılan local lineage karşılaştırması
- Kapsam:
  - statusline ya da statusline-capable ilgisi olan 24 public GitHub projesi
  - lineage analizi için 2 public gist
  - repo stars, push tarihleri, release tag'leri ve install yolları yayın gününde canlı olarak yeniden kontrol edildi

## Ana Anlatı Dışında Kalan Önemli Projeler

pcvelz/ccstatusline-usage (<https://github.com/pcvelz/ccstatusline-usage>)

FlineDev/CustomStatusline (<https://github.com/FlineDev/CustomStatusline>)

ohugonnot/claude-code-statusline (<https://github.com/ohugonnot/claude-code-statusline>)

xleddyl/claude-watch (<https://github.com/xleddyl/claude-watch>)

syou6162/ccstatusline (<https://github.com/syou6162/ccstatusline>)

daliovic/cc-statusline (<https://github.com/daliovic/cc-statusline>)

rz1989s/claude-code-statusline (<https://github.com/rz1989s/claude-code-statusline>)

Wangnov/claude-code-statusline-pro (<https://github.com/Wangnov/claude-code-statusline-pro>)

sotayamashita/claude-code-statusline (<https://github.com/sotayamashita/claude-code-statusline>)

ryoppippi/ccusage (<https://github.com/ryoppippi/ccusage>)

leeguooooo/claude-code-usage-bar (<https://github.com/leeguooooo/claude-code-usage-bar>)

ccstatusline (<https://github.com/sirmalloc/ccstatusline>)

`/oauth/usage`

`npx ccusage statusline`

aude-code-usage-bar)

chongdashu/cc-statusline (<https://github.com/chongdashu/cc-statusline>)

## Dikkat Çeken Gist'ler

SippieCup/Ocd2567... (<https://gist.github.com/SippieCup/Ocd256789b6350196fc34c6d0ac09871>)

jtbr/4f99671... (<https://gist.github.com/jtbr/4f99671d1cee06b44106456958caba8b>)

## Sonuç

Claude Code statusline pazarı artık o kadar olgun ki "en iyisi hangisi?" sorusu tek başına doğru soru değil. Daha doğru soru şu: hangi çalışma tarzı için en iyisi?

- Çoğu kullanıcı için: **ccstatusline** (<https://github.com/sirmalloc/ccstatusline>)
- Plugin-first şıklık için: **claude-powerline** (<https://github.com/Owloops/claude-powerline>)
- Starship ağırlıklı Rust kullanıcıları için: **cship** (<https://github.com/stephenleo/cship>)
- Gözlem ve görünürlük için: **claude-hud** (<https://github.com/jarrodwatts/claude-hud>)
- Daha muhafazakâr operatörler için: **claudeline** (<https://github.com/fredrikaverpil/claudeline>)

- Shell seven ve özellik sayısından çok tek bakışta görsel tatmin isteyenler için: [SippieCup](https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871) (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>) çizgisindeki Ghostty dostu varyantlar

Son cümleyi özellikle bırakmak istiyorum. En tatmin edici statusline'lar her zaman en uzun README'ye ya da en fazla widget'a sahip olanlar olmuyor. Bazen kazanan, neredeyse okumaya gerek kalmadan anlayabildiğin çizgi oluyor.

---

## Bu Araştırma Hakkında

Bu analiz, ajan altyapısı odaklı sürekli bir araştırma akışının parçası olarak hazırlandı. Destekleyici kanıtlar ve dış bağlantılar raporla birlikte sürümlendiği için gerekçe zaman içinde denetlenebilir kalır.

Araştırma Yiğit Konur • April 16, 2026 • [metodoloji](#)

---

### KAYNAKLAR

- [1] Claude Code statusline docs (<https://docs.anthropic.com/en/docs/claude-code/statusline>)
- [2] anthropics/claude-code statusline issues (<https://github.com/anthropics/claude-code/issues?q=is%3Aissue+statusline>)
- [3] sirmalloc/ccstatusline (<https://github.com/sirmalloc/ccstatusline>)
- [4] Haleclipse/CCometixLine (<https://github.com/Haleclipse/CCometixLine>)
- [5] Owloops/claude-powerline (<https://github.com/Owloops/claude-powerline>)
- [6] jarrodwatts/claude-hud (<https://github.com/jarrodwatts/claude-hud>)
- [7] stephenleo/cship (<https://github.com/stephenleo/cship>)
- [8] fredrikaverpil/claodeline (<https://github.com/fredrikaverpil/claodeline>)
- [9] kamranahmedse/claude-statusline (<https://github.com/kamranahmedse/claude-statusline>)
- [10] daniel3303/ClaudeCodeStatusLine (<https://github.com/daniel3303/ClaudeCodeStatusLine>)
- [11] felipeelias/claude-statusline (<https://github.com/felipeelias/claude-statusline>)
- [12] Astro-Han/claude-pace (<https://github.com/Astro-Han/claude-pace>)
- [13] wolfdendepublishing/pyccsl (<https://github.com/wolfdendepublishing/pyccsl>)
- [14] kcchien/claude-code-statusline (<https://github.com/kcchien/claude-code-statusline>)

-

- [15] chongdashu/cc-statusline (<https://github.com/chongdashu/cc-statusline>)
- [16] leeguooooo/claude-code-usage-bar (<https://github.com/leeguooooo/claude-code-usage-bar>)
- [17] ryoppippi/ccusage (<https://github.com/ryoppippi/ccusage>)
- [18] SippieCup statusline gist (<https://gist.github.com/SippieCup/0cd256789b6350196fc34c6d0ac09871>)
- [19] jtbr statusline gist (<https://gist.github.com/jtbr/4f99671d1cee06b44106456958caba8b>)