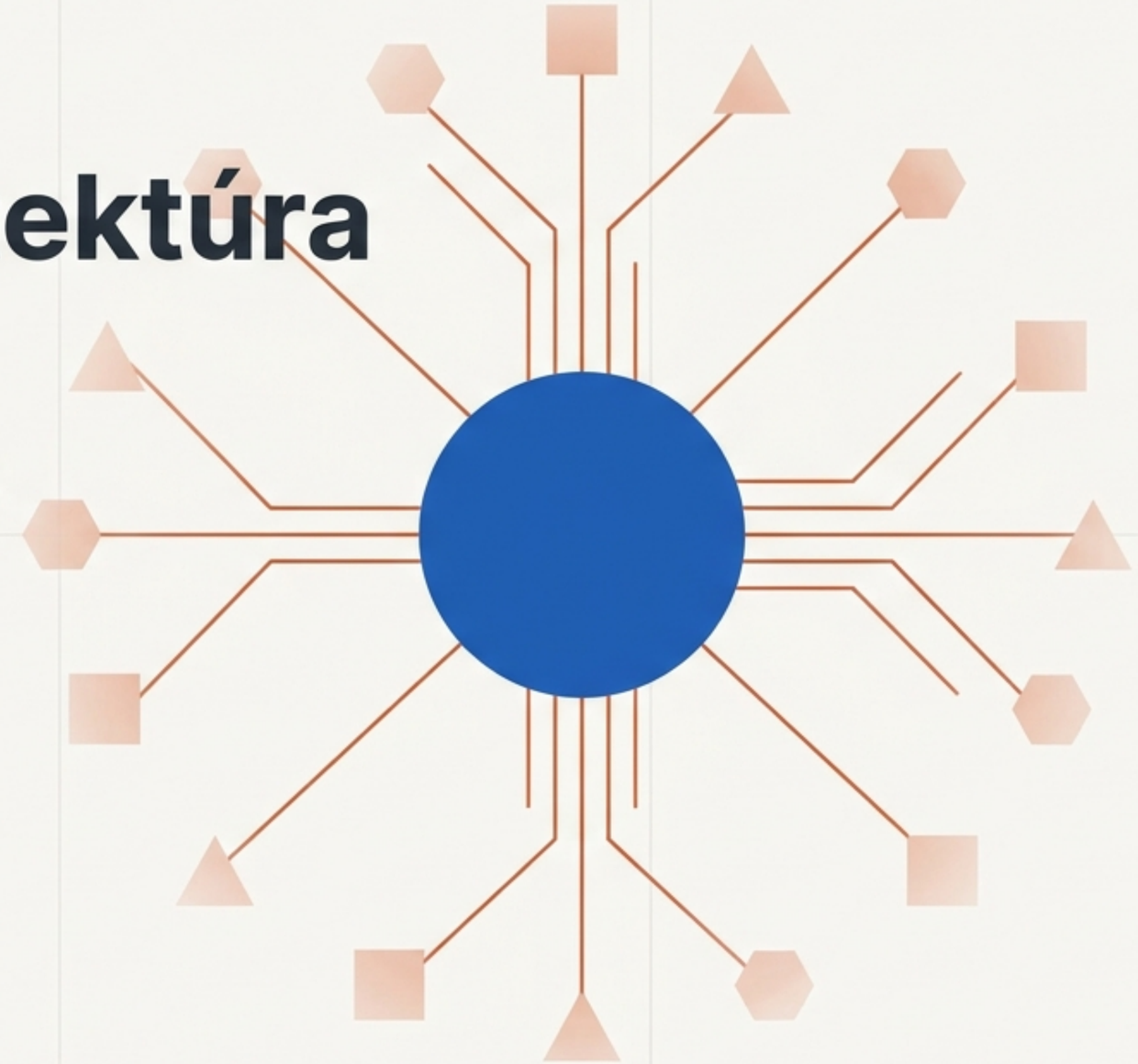
Az ügynök-architektúra újragondolása

Miért téves a több alügynök = jobb rendszer mítosz, és hogyan optimalizáljuk a kódoló AI-t?

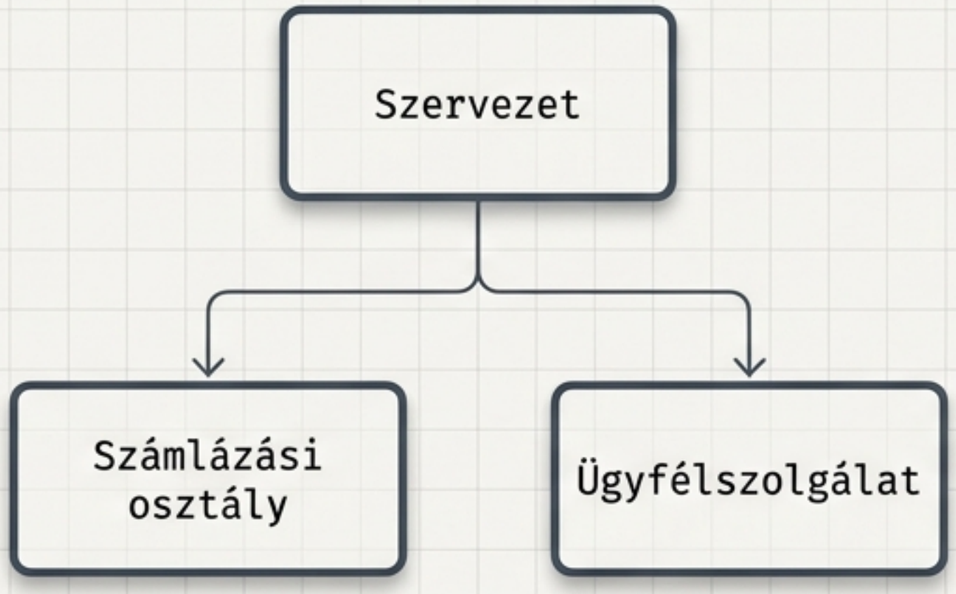


Alapozva Diane Sarcis (2026. október 22.) iránymutató elemzésére és a Panda Code gyakorlati alkalmazására.

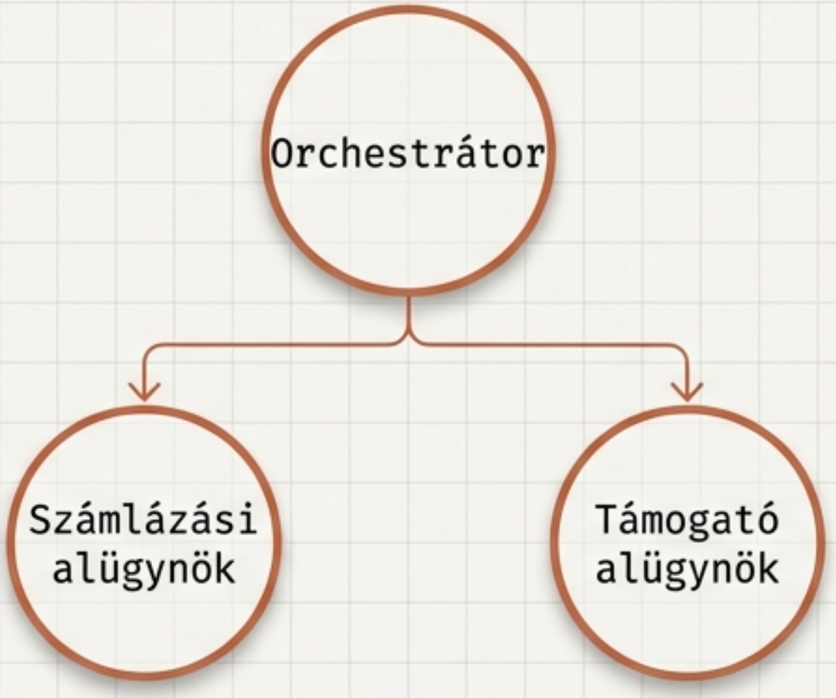
Gemini Notebook

A Conway-törvény csapdája az AI-ban

A szervezeti felépítés



Az AI architektúra

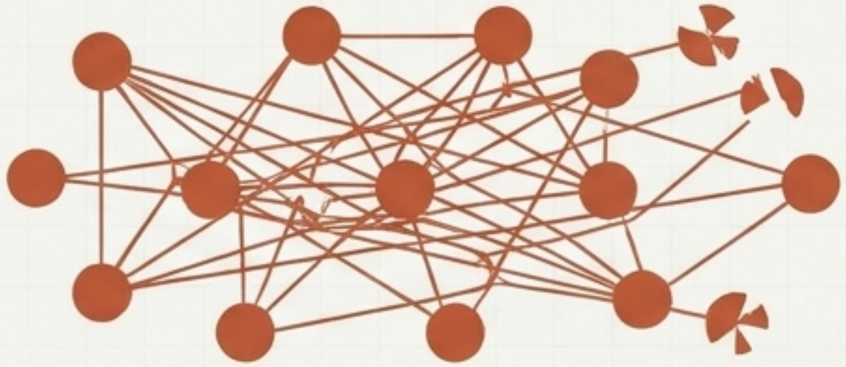


A cégek hajlamosak a saját szervezeti ábrájukat lemásolni az ügynökök tervezésekor. Papíron logikusnak tűnik, de a valóságban minőségromlást, magasabb költségeket és megnövekedett késleltetést (latency) eredményez. A feladatok túlzott szétbontása nem az AI logikáját, hanem az emberi csapatok korlátait tükrözi.

Gemini Notebook

A túlburjánzó architektúra 4 tünete

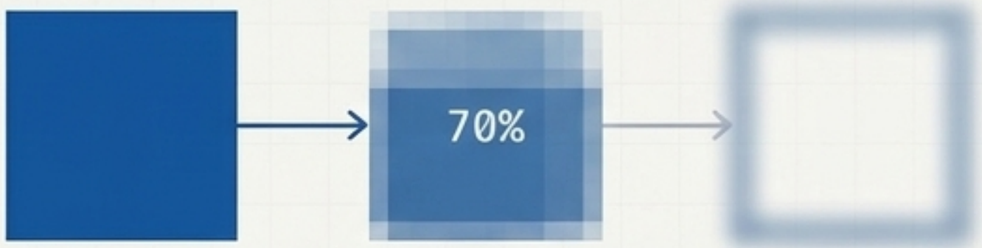
1. Túlburjánzás



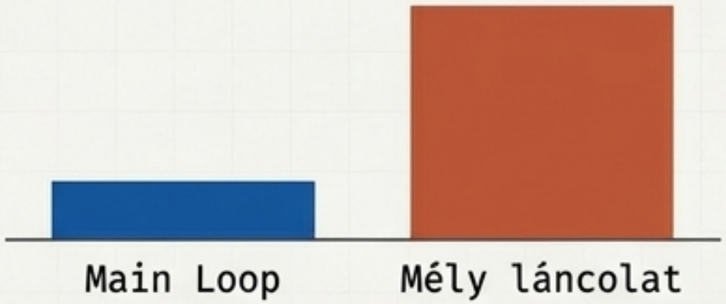
2. Franken-prompt

```
prompt = """
  Válaszolj magyarul.
  A válaszban ne használj magyar szavakat.  ← [ELLENTMONDÁS]
  Fogalmazz tömören.
  Fejtsd ki bővebben.                       ← [ELLENTMONDÁS]
  Légy kreatív.
  Tartsd magad a tényekhez.                 ← [ELLENTMONDÁS]
"""
```

3. Telefonjáték-effektus



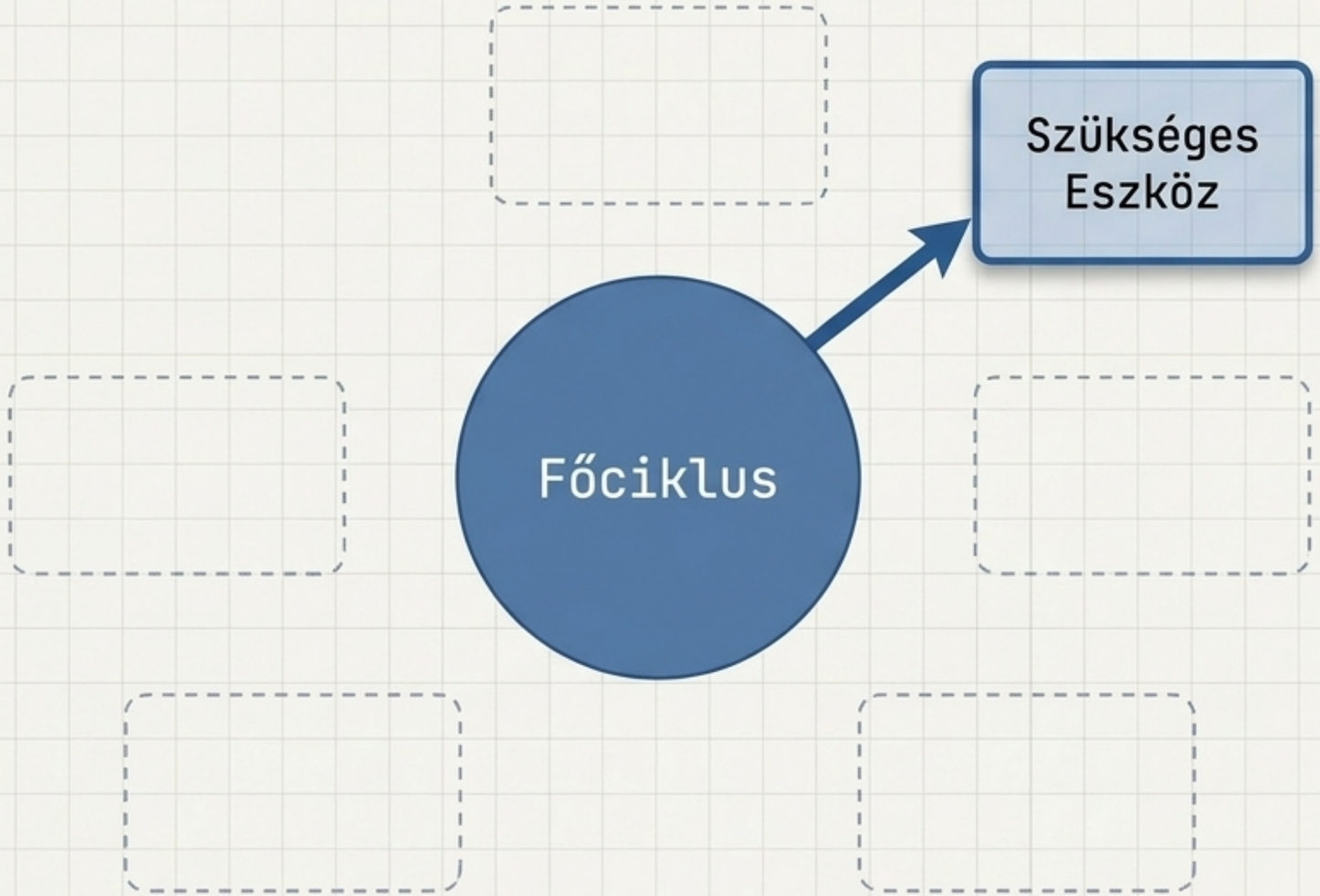
4. Késleltetés



Egyetlen ciklus 2x gyorsabb a mély láncolatnál.

Gemini Notebook

A megoldás: Progresszív Felfedés (Progressive Disclosure)



Nincs szükség minden szabály és eszköz betöltésére minden egyes futtatáskor. Az ügynök csak a rövid leírásokkal indul, és a teljes eszközt/készséget csak a tényleges szükség pillanatában hívja be.

Siker: Az Anthropic ezzel a módszerrel **több mint 80%-kal csökkentette a Claude Code rendszer-promptját** a fejlettebb modelleknél.

Gemini Notebook

Mi hova kerül? Az egyciklusos topológia alkotóelemei



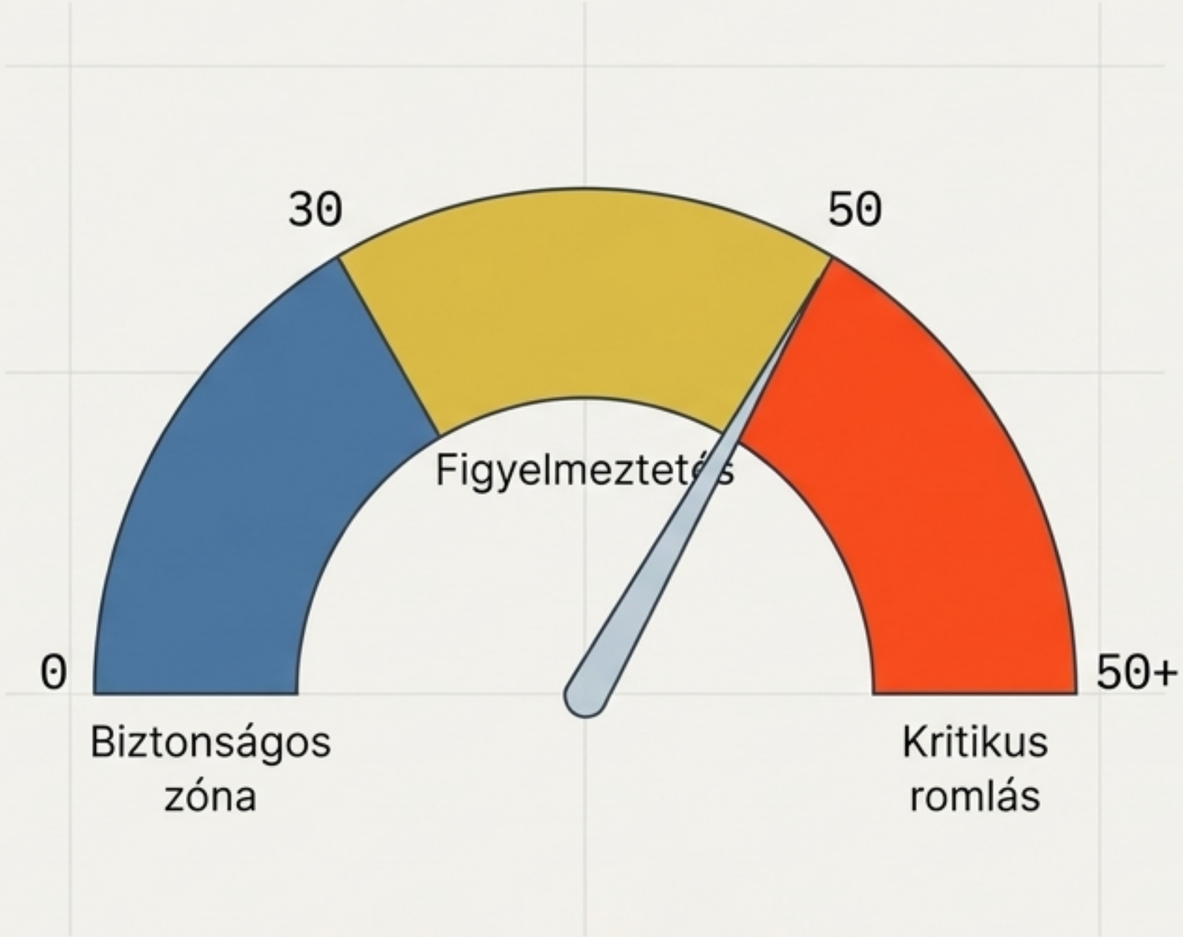
[SKILL]	**Szakterületi mérlegelés** (Domain Judgment) – pl. hogyan kezeljen egy speciális esetet az AI.
[TOOL]	**Cselekvés (Action)** – élő adat lekérése külső rendszerekből, szigorú korlátok betartása.
[LOOKUP]	**Ritka tények (Occasional Facts)** – tudás, amely ritkán kell, és nem való az állandó promptba.

Egyetlen jól strukturált ciklus, amely megfelelő tárolókba helyezi a tudást, megtisztítja a legkaotikusabb rendszereket is.

Gemini Notebook

A beolvasztás korlátja: A 30-50 eszközös határ

Az összevonás nem jelenti a prompt esztelen telezsúfolását. Ha az eszközdefiníciók száma meghaladja a 30-50 darabot, a modell eszközválasztási (tool choice) képessége drasztikusan romlani kezd.



A Szabály:
Kis számú, jól elnevezett eszközcsoportokat használjunk a felhasználói szándék alapján, a többit pedig csak progresszíven fedjük fel.

Gemini Notebook

A biztonságos átállás iteratív ciklusa

1. Egyetlen alügynök beolvasztása

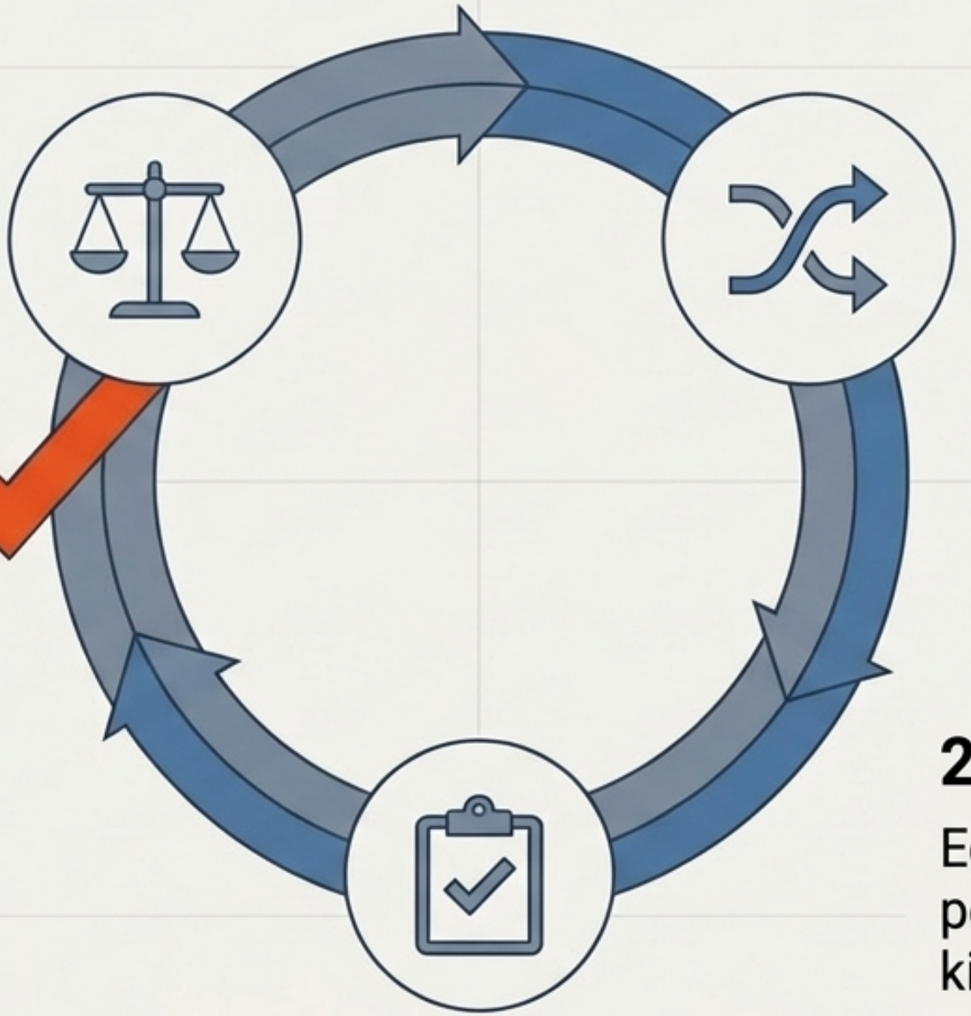
Kezd a leggyakrabban használt párral.
Prompt -> Skill, API -> Tool.

2. A 10 Arany Példa

Egy izolált tesztkészlet, ahol pontosan ismert a tökéletes kimenet.

3. Többszörös mérés

A régi és az új struktúra alapos összehasonlítása.



FIGYELEM: Minden kiértékelést futtass le többször!
Az összevonás néha csendben töri meg a folyamatot. Ha csak egyszer tesztelsz, egy érmefeldobást juttatsz éles üzembe.

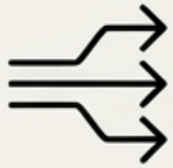
Gemini Notebook

A kivételek: Mikor nyernek az alügynökök?



Független Ellenőrzés

Friss kontextusú ügynök ellenőrzi a munkát, mert a készítő ügynök hajlamos megmagyarázni a saját hibáit.



Párhuzamos Munka

Egymástól független kutatási szálak, amelyek tiszta, eldobható kontextusban futnak egyszerre.



Moduláris Jogosultságok

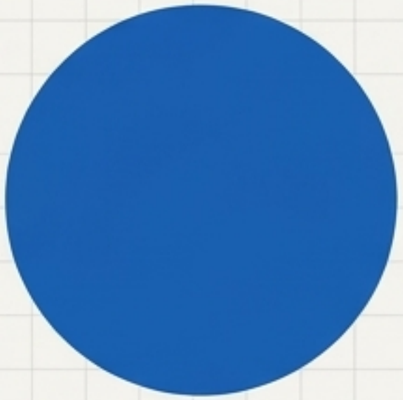
Amikor a szűk feladathoz csak minimális, szigorúan korlátozott hozzáférésre van szükség.

AZ ARANY SZABÁLY: A szétválasztás alapja mindig a kontextus, sohasem a probléma típusa. A különálló ügynöknek tiszta, kompakt eredményt kell visszaadnia.

Gemini Notebook

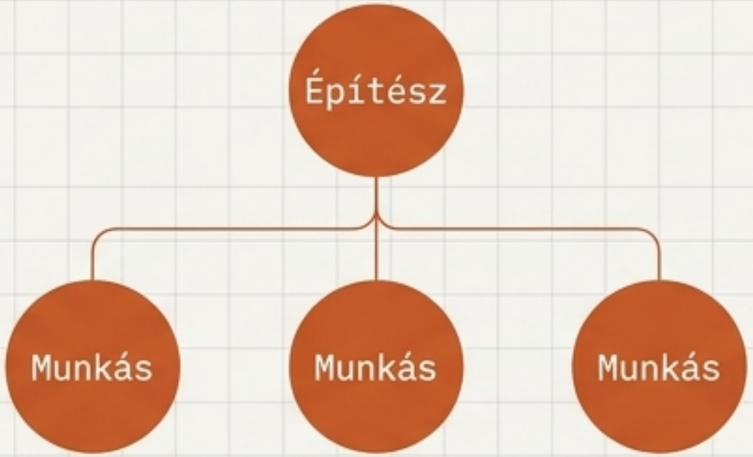
Panda Code a gyakorlatban: Elméletből topológia

Egyetlen Chat (Single Chat)



- Nincs delegálás
- Nincs kontextus-vesztés
- Azonnali, interaktív szerkesztés

Csapat Mód (Team Mode)

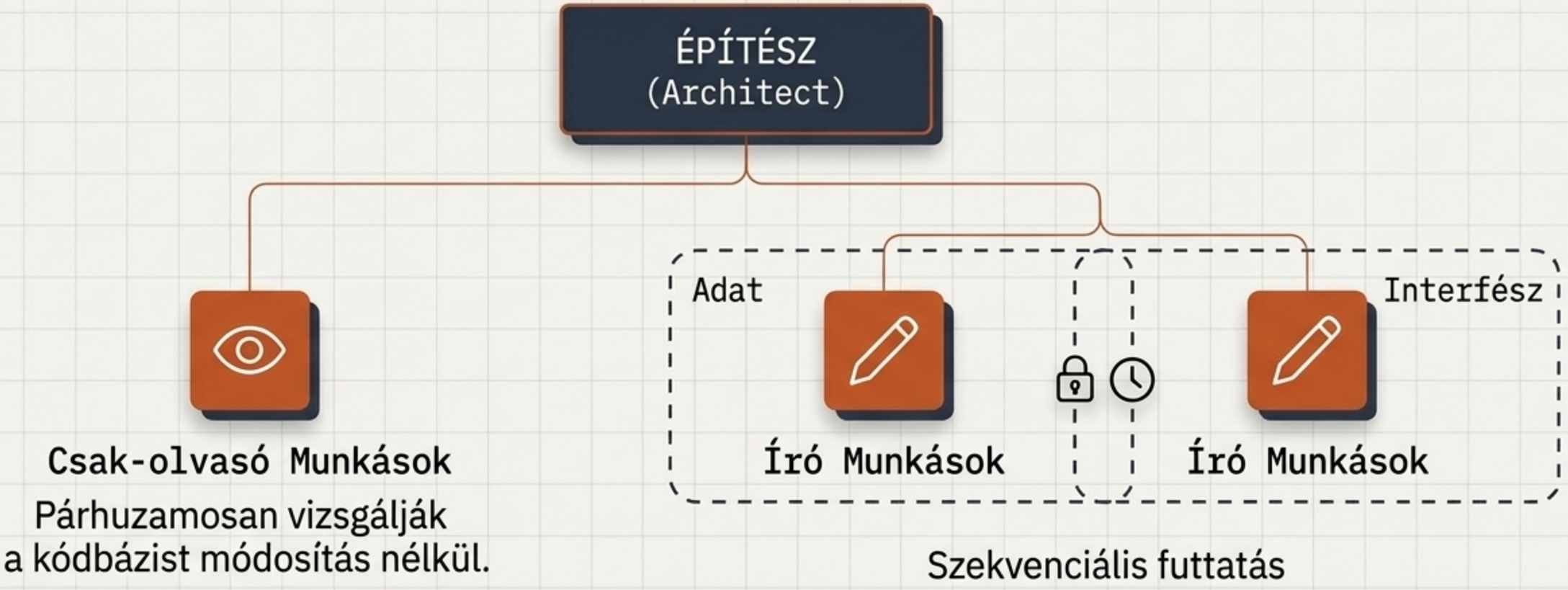


- Párhuzamos végrehajtás
- Izolált, friss kontextus
- Az Építész delegál és ellenőriz

A Panda Code pontosan a cikk által felvázolt strukturális választási lehetőséget kínálja egyetlen gombnyomással.

Gemini Notebook

A Csapat-topológia anatómiája a Panda Code-ban



A hatókörök (file scopes) biztosítják, hogy az egy gépen lévő kódon dolgozó munkások ne írják felül egymás munkáját. Ha a hatókörök ütköznek, a rendszer párhuzamos helyett egymás után futtatja őket.

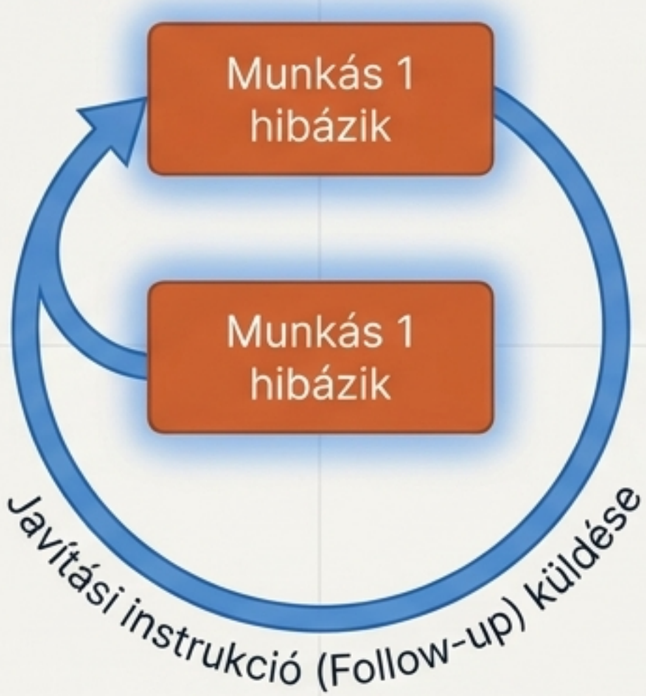
Gemini Notebook

A Follow-up mechanika: A kontextus megmentése

Hagyományos Handoff



Panda Code Follow-up



A Munkás megőrzi a megszerzett **tudást** a kódról.

Drágább egy teljesen új munkásnak nulláról elmagyarázni a kontextust, mint visszaküldeni a javítást ugyanannak a csomópontnak. Ez a mechanika egy elkerült handoff, amely drasztikusan csökkenti a hibarátát.

Gemini Notebook

Architekturális Trade-off Mátrix

	FŐCIKLUS / EGYETLEN CHAT	ALÜGYNÖKÖK / CSAPAT MÓD
Költségprofil	Tokentakarékos (Egyetlen futás)	Gyors tokenégetés (Több modell, párhuzamos futás)
Kontextus integritás	Teljes szál, megosztott, folyamatos állapot	Telefonjáték veszély, tömörített összefoglalók a handoff miatt
Ideális felhasználás	Elírások javítása, kis konfig cserék, interaktív lépésről-lépésre hibakeresés.	Párhuzamos kutatás, tömeges szerkesztés, kontextusablakot kitöltő óriás feladatok.

A Csapat Mód nem ingyen sebesség. Párhuzamos időt és friss szemeket vásárolsz, de tokenekkel és enyhe adatvesztéssel fizetsz érte.

Gemini Notebook

Az Ügynök-Architektúra Döntési Fája



Ha nem tudod pontosan megnevezni az okot, maradj a főciklusnál. Az esetek többségében ez a gyorsabb és olcsóbb meg

Gemini Notebook