RX Event Taxonomy – White Venom

Ez a dokumentum **puska** és **belső referencia**.
Nem oktatóanyag, nem marketing, nem kompromisszum.
A rendszer *reaktív idegrendszerének* taxonómiája.

Alapelv

Az **event nem adat**, hanem **jelentés**.

Az RX modellben az event: - nem dönt - nem hajt végre - nem ismeri a következményeket

Csak **közöl**.

A reakció: - Observer - Scheduler - Executor

Ezért az eventeket **osztályozni kell**, különben a rendszer: - összefolyik - vak lesz - instabillá válik

---

Event dimenziók (tengelyek)

Egy esemény **nem egy dimenziós**.
Legalább **négy ortogonális tengelye van**.

---

1. Origin – Eredet

Honnan származik az esemény?

Ez határozza meg: - ki reagálhat rá - milyen jogkörrel

Origin kategóriák

- `BOOTSTRAP`
- rendszerindítás
- init invariánsok

- egyszeri, megismételhetetlen

- `RUNTIME`

- futás közbeni állapot

• élő rendszer

• `SYSTEM_STATE`

• kernel, fs, mem, time

• belső állapotváltozás

• `USER_INTENT`

• admin / fejlesztői szándék

• explicit, auditálható

• `SECURITY_SIGNAL`

• audit
• integrity

• policy megsértés

• `EXTERNAL_INPUT`

• hálózat
• időforrás
• külső trigger

---

2. Temporal Nature – Időviszony

Hogyan viszonyul az időhöz?

Ez **scheduler-kérdés**, nem busz-logika.

Időosztályok

• `INSTANT`
• egyszeri esemény

• pl. boot flag

• `CONTINUOUS`

• folyamatos állapot

• pl. time drift, sysctl eltérés

- `PERIODIC`

- ütemezett

- pl. integrity check

- `BURST`

- eseménycsúcs
- pl. audit spike, attack pattern

---

3. Severity – Biztonsági súly

Nem log level.
Ez **rendszerkockázat**.

Súlyszintek

- `INFORMATIONAL`
- megfigyelés

- nincs beavatkozás

- `DEVIATION`

- eltérés

- még nem sért

- `POLICY_VIOLATION`

- szabálysértés

- reakció szükséges

- `INTEGRITY_BREAK`

- invariáns sérült

- fail-hard lehet

- `COMPROMISE_INDICATOR`

- feltételezett betörés
- izoláció / leállítás

---

4. Reaction Class – Reakció típusa

Mit vár el a rendszer válaszként?

Ez **nem konkrét művelet**, hanem *reakciós osztály*.

Reakciók

- `OBSERVE_ONLY`
- log
- metrika
- `STATE_UPDATE`
- belső állapot frissítése
- `ENFORCEMENT_REQUIRED`
- policy kényszerítése
- `SYSTEM_TRANSITION`
- állapotváltás
- pl. lockdown
- `TERMINATE_CONTEXT`
- folyamat / kontextus megszüntetése

Példa – Runtime sysctl eltérés

```
Origin: SYSTEM_STATE
Temporal: CONTINUOUS
Severity: POLICY_VIOLATION
Reaction: ENFORCEMENT_REQUIRED
```

Értelmezés: - az esemény terjed a Bus-on - Observer felismeri - Scheduler prioritást ad - Executor kényszerít

Az event **nem hajt végre**.

Hot vs Cold Observables

Cold Observable

- bootstrap
- init konfiguráció
- invariánsok

Tulajdonság: - új subscriber → újrafut

Hot Observable

- runtime drift
- audit események
- ptrace attempt
- mount változás

Tulajdonság: - folyamatos - nem replayelhető

Backpressure – stabilitási modell

A backpressure **nem teljesítmény kérdés**.

Security környezetben: - burst ≠ több thread - zaj ≠ több log

RX eszközök: - buffer - window - sample - aggregation

Cél: - rendszer stabil maradjon - jel ne vesszen el

Ami NEM része a taxonómiának

❌konkrét parancs
❌file path
❌bash logika
❌végrehajtási részletek

Ezek az **Observer / Module / Executor** felelősségei.

Záró axióma

> **Az event azt mondja meg, mi történt.**
> **A rendszer dönti el, mi legyen belőle.**

Ez a White Venom RX idegrendszerének alapja.