

# MCAW 2.0 – Scénářové simulace (offline)

Tento dokument je \*\*produktový i ladicí\*\*:

- pro Product Ownera: definuje scénáře a otevření (co má aplikace dělat)
- pro ladění: obsahuje přesně to, co engine používá (prahy, guardy, hysterese), aby výstup reportu byl zpřístupnitelný auditovatelný

> Pozn.: scénáře jsou navrženy tak, aby byly \*\*deterministické\*\* a běžely v CI (bez videí / bez cloutu). Reálné video replay je samostatná vrstva.

---

## 1) Co se testuje

Scénáře validují chování `RiskEngine.evaluate()` (MCAW 2.0):

- dosažitelnost ORANGE/RED v kritických situacích
- stabilita přechodů (hystereze, bez cvakání)
- robustnost při nízké kvalitě obrazu (qualityWeight)
- ROI edge / off-center chování (parked cars, protijedoucí mimo koridor)
- moto specifika (leanDeg, jitter, menší cíle)
- brake cue jako zesilovač v nebezpečných situacích

Výstup je \*\*dual-use\*\*:

- 1) \*\*Markdown report\*\* (lidsky čitelný)
- 2) \*\*JSONL event log\*\* (strojově čitelný pro regresní ladění)

Reporty se generují vždy, i při PASS.

---

## 2) Prahy a guardy píevzaté z kódu (source of truth)

### 2.1 Mode thresholds (`RiskEngine.thresholdsForMode(mode)`)

effectiveMode	popis	ttcOrange	ttcRed	distOrange	distRed	relOrange	relRed
--- --- --- --- --- --- --- ---							
1   Místo (default)	3.0s   1.2s   15m   8m   3 m/s   5 m/s						
2   Sport / dálnice	4.0s   1.5s   30m   12m   5 m/s   9 m/s						
3   Uživatel	z `AppPreferences`   z `AppPreferences`   z `AppPreferences`   z `AppPreferences`   z `AppPreferences`						

### 2.2 Risk hysteresis (`RiskEngine.riskToLevelWithHysteresis(risk, conserv)`)

Quality (qW) ovlivňuje `conserv = 1 - clamp(qW, 0.60..1.0)`.

- `orangeOn = 0.45 + 0.17 \* conserv`
- `redOn = 0.75 + 0.07 \* conserv`
- `orangeOff = orangeOn - 0.06`

- `redOff = redOn - 0.05`

### 2.3 RED combo guard (`allowRed`)

RED (level=2) je povolen jen pokud:

```
allowRed = strongTtc && (
strongDist || strongRel ||
(slopeStrong && (midDist || midRel))
)

slopeStrong: ttcSlopeSecPerSec <= slopeThr
slopeThr = -1.0 - 0.40*conserv

strongK = 0.85 + 0.05*conserv
midK = 0.60 + 0.10*conserv
```

To je klíčové pro audit: report musí uvádět, zda by RED prošel nebo byl guardován.

---

### 3) Formát scénáře (vstupy)

Scénář je složen ze segmentů. Každý segment generuje masovou adresu vstupu pro `RiskEngine.evaluate()`:

- distanceM (m)
- approachSpeedMps (m/s)
- ttcSec (s)
- ttcSlopeSecPerSec (s/s)
- roiContainment (0..1)
- egoOffsetN (0..2)
- cutInActive (bool)
- brakeCueActive + brakeCueStrength (0..1)
- qualityWeight (0.60..1.0)
- riderSpeedMps (m/s)
- egoBrakingConfidence (0..1)
- leanDeg (deg; NaN pro auto)

Scénář navíc definuje \*\*ořekávání\*\* (regresní kontrakty), typicky:

- do kdy má nastat ORANGE/RED po hazard momentu
- co se nesmí stát (např. alert pokud invalid TTC + rel≈0)
- limit počechodů v masovém okně (anti-flap)

---

### 4) Formát výstupu reportu (co musí obsahovat)

### 4.1 Markdown report (PO + ladění)

- Story

- effective config
- derived thresholds (přesně z kódu)
- expectations + PASS/FAIL per rule
- tabulka klíčových přechodů (ALERT\_ENTER/ALERT\_EXIT)
- t, level, risk, reasonShort + reasonBits
- vstupy v daném okamžiku (dist/rel/ttc/slope/roi/qW)

#### ### 4.2 JSONL event log

- každý event jako JSON rámek
- vždy obsahuje derived thresholds (orangeOn/redOn/slopeThr/strongK/midK)
- extra pole (dist/rel/ttc/roi/qW/segment)

To umožní: grep, diff, regresní sledování a rychlé určení "proč nebyl RED".

---

## 5) Katalog scénářů (aktuální)

Katalog je implementován v `ScenarioCatalogFactory` a bývá v testu `ScenarioSimulationReportTest`.

#### ### City (auto)

- \*\*C1\_CITY\_PARKED\_PASS\_BY\*\* – parked cars na kraji ROI (žádné alerty)
- \*\*C2\_CITY\_JAM\_APPROACH\*\* – dojíždění do kolony (ORANGE→RED)

#### ### Tunnel (auto)

- \*\*T1\_TUNNEL\_EXPOSURE\_DROP\*\* – quality drop v tunelu + pokračující closing (stabilita + RED pokud hazard trvá)

#### ### Highway (auto)

- \*\*H1\_HIGHWAY\_STEADY\_FOLLOW\*\* – stabilní odstup, bez falešných alarmů
- \*\*H2\_HIGHWAY\_SUDDEN\_BRAKE\*\* – náhlé brzdění lead car (ORANGE→RED; brakeCue active)

#### ### Rural (auto)

- \*\*R1\_RURAL\_CURVE\_ONCOMING\*\* – protijedoucí mimo ROI/koridor (žádné alerty)

#### ### Moto

- \*\*M1\_MOTO\_FOLLOW\_CURVE\*\* – motorka před motorkou v zatáčce (lean, stabilita, ORANGE při closing)
- \*\*M2\_MOTO\_JAM\_SUDDEN\_BRAKE\*\* – náhlé brzdění před motorkářem (ORANGE→RED)

---

## 6) Kde najít reporty

Po spuštění unit testů:

- `app/build/reports/mcaw\_scenarios//INDEX.md`
- `app/build/reports/mcaw\_scenarios//.md`

- `app/build/reports/mcaw\_scenarios//.jsonl`

Tyto reporty jsou určené pro iterativní ladění a zároveň pro PO verifikaci odkládání.