

Formal Methods at Cloud Scale: Before, During, and After AI

Claudia Cauli

Principal Research Engineer & Team Lead @ Huawei Ireland RC

Who am I

9+ years doing R&D in Cloud companies during AI scrambles



PhD in Logic for Computer Science University of Gothenburg



Applied Scientist AWS Automated Reasoning Group



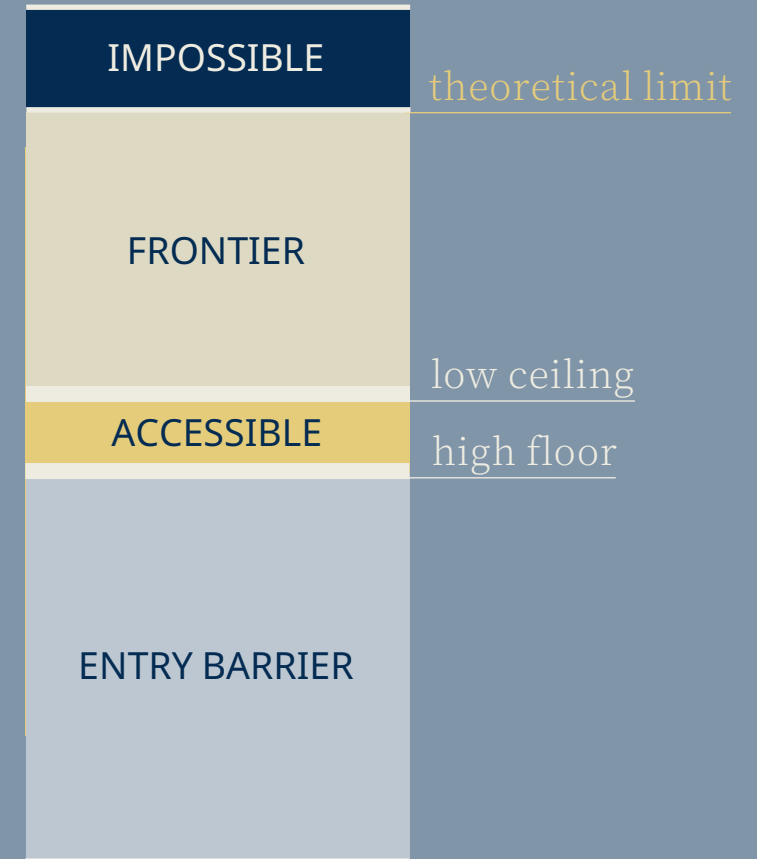
Principal Research Engineer & Team Lead Huawei Cloud

Keywords Correctness · Reliability · Security · Cloud · Code · Configuration

Did FM just get cheap?

Act I: Before

Limited FM reach



What FM grabbed

- **Protocol Modeling** Highest-stake infra
key-value stores and databases (*Huawei Cloud K2, AWS, MongoDB, Azure CosmosDB, etc.*) ·
load balancers (*Huawei Cloud GSLB*) · storage (*AWS S3*)
- **Code Verification** Critical small libraries, compilers, kernels
task sharder (*Huawei Cloud GSLB*) · crypto libraries (*AWS s2n-bignum*) · authorization
engine (*AWS IAM*) · Linux eBPF · DPDK eBPF (*HW Cloud*) · seL4 · CompCert
- **Configuration Safety** SMT-based cloud customer analyses:
privilege escalation (*Huawei Cloud IAM*) · Access Analyzer (*AWS IAM*) · VPC Analyzer (*AWS*) ·
network verification (*Alibaba Cloud, Google*)

What FM left to SRE Ops

- **Resilience & availability**
- Large-scale prod incidents
- SRE, Ops, DevOps
- Generic code quality / **QA**
- SRE folks side-eye FM
- **War-rooms**

Known unknowns Predictable, latent, need hunting (TLA+/P modeling, verification, PBT, determ. simulation testing)	Unknown unknowns Unpredictable, sudden, slow (monitoring, chaos engineering, deterministic simulation)
Known knowns Predictable, sudden, violent (ops engineering, redundancy, failover, runbooks, SOPs)	Unknown knowns Unpredictable, transient, slow (metastability mapping, dependency tracing)

SRE Rumsfeld matrix

The FM improv

Yes,

Proofs for critical systems

High cost of incorrectness, higher **ROI**

Sound bug finding

Deals with **known unknowns**

...but

Niche expertise

High **entry barrier**

Limited **tools usability**

Domain knowledge bottleneck

The FM improv

Yes,

Proofs for critical systems

High cost of incorrectness, higher **ROI**

Sound bug finding

Deals with **known unknowns**

...and

Niche expertise

High **entry barrier**

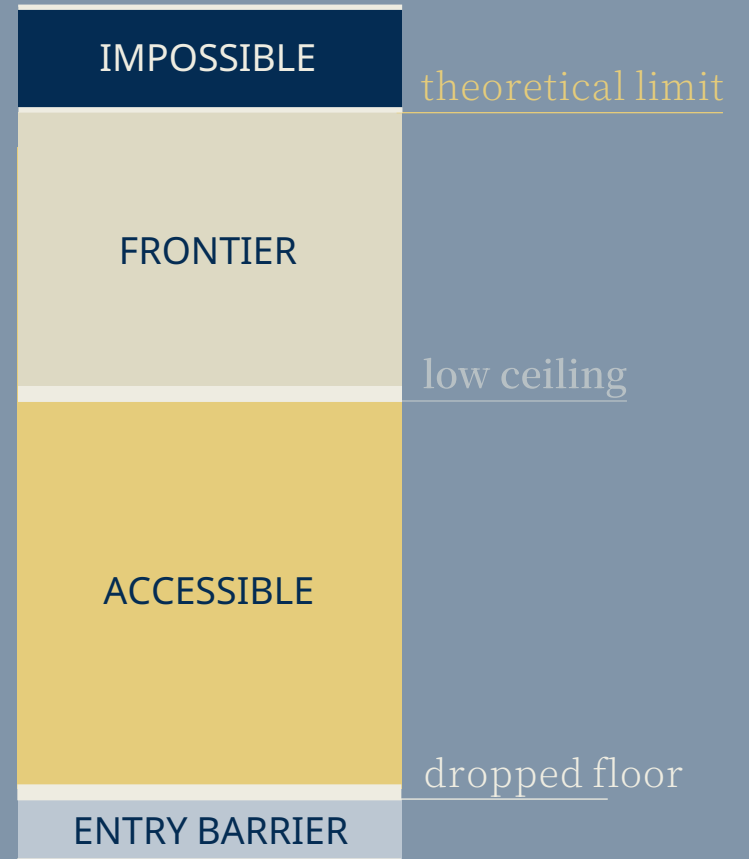
Limited **tools usability**

Domain knowledge bottleneck

these just collapsed

Act II: During

The floor has dropped



The floor has dropped

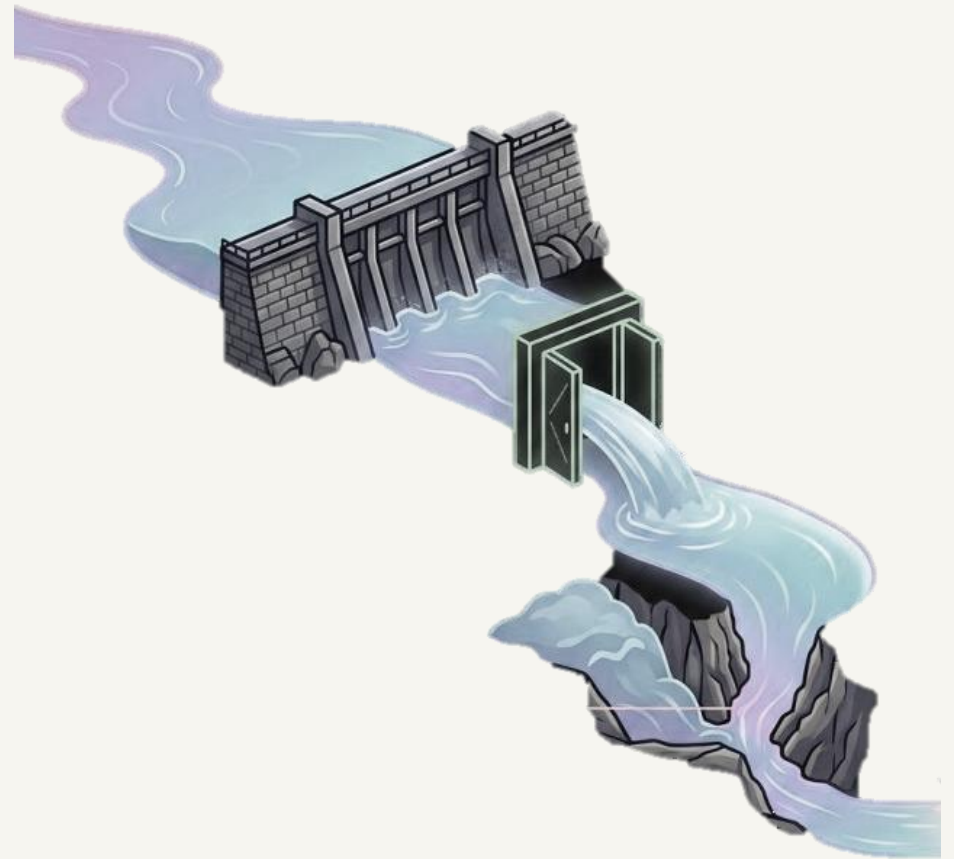
Knowledge bottleneck to burst dam

Expertise reproduced at machine-scale

Validation constraint *moved* downstream

From person-months to *person-days*

Making FM *indispensable* in this new SDL



I'm sure we're all working on

NL2SMT autoformalization, LLM + solver loop, red-teaming, consistency

SpecAgent purpose-built agent driving requirement generation

VeriAgent purpose-built driving escalating-strength FM tools

Code2Model automated TLA+ model extraction from code (Specula-like)

DPDK eBPF general-purpose code agents driving CBMC, ESBMC, Frama-C/WP

- *hours* of find bug → fix → verify fix → repeat
- *days* of human triage
- hit the under-spec'd trap
- hit symbolic tools *practical limits* (e.g. SMT choking on bit-vector multiplication)

Blind spots

Also **amplified**, becoming even more dangerous than before:

Intent ambiguities

Validation, **vacuity**, under-specified systems

Plausible-looking outputs, shallowly reviewed

Either we *constrain AI* or we meticulously *automatically guard blind spots*, **or both**

Lifting our role to a higher abstraction layer

The *theory's* the limit

AI won't make the undecidable decidable

AI can help lift **practical ceilings**, like

LLM-guided *SMT heuristics*

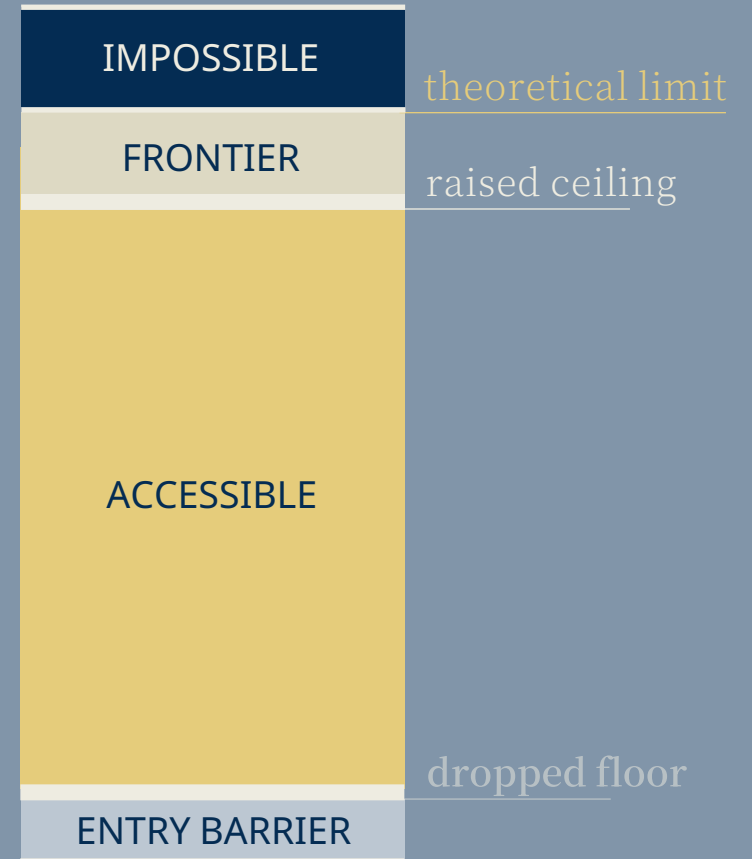
LLM-guided *invariant* generation

LLM-guided *harness* generation

LLM-guided *proof* generation

Act III: After

The ceiling will shoot up



Neurosymbolic > Neuro

- LLMs are **resource-wasteful**, unfocused / broadly scoped, and **unsound**
- Symbolic tools are **resource-optimized**, focused / narrowly scoped, and **sound**
- On reasoning tasks, neurosymbolic is likely to be more **resource-effective** and **trustworthy**
- We need **data**, task parity benchmarks, proper evals
- Time, tokens → CPU/GPU cycles → electricity → money

Given comparable outputs, how much does each cost?

Faithful Autoformalization

- **Crime-tier** & **cost-tier** risks live where customer **contracts** meet internal **implementation**.
 - regulatory compliance
 - data compliance
 - billing regulations
- We need faithful **bi-directional (BX)** autoformalization capability

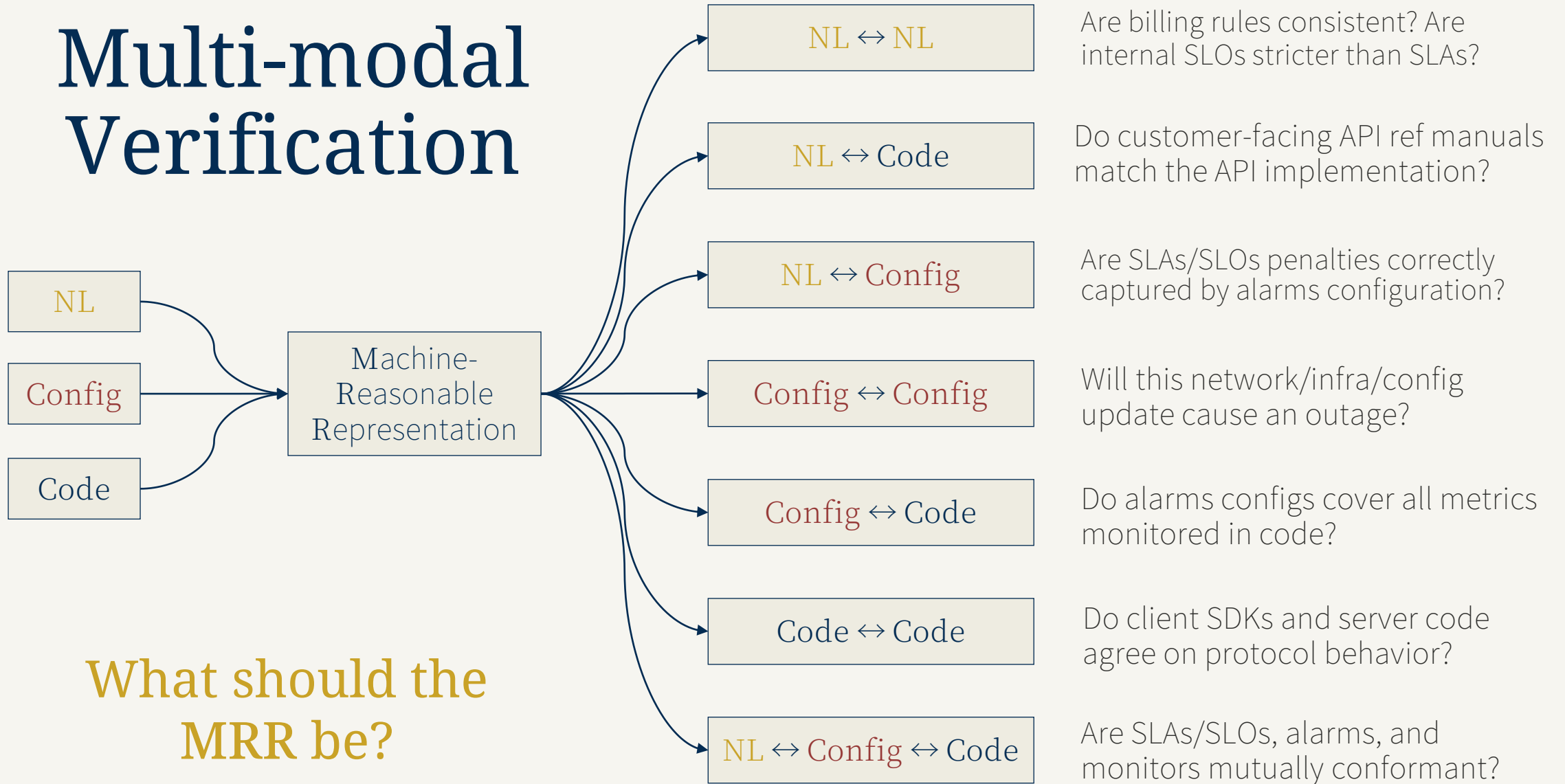
NL ↔ NL

NL ↔ Code

NL ↔ Config

How do we know it is faithful? Faithful to what?

Multi-modal Verification



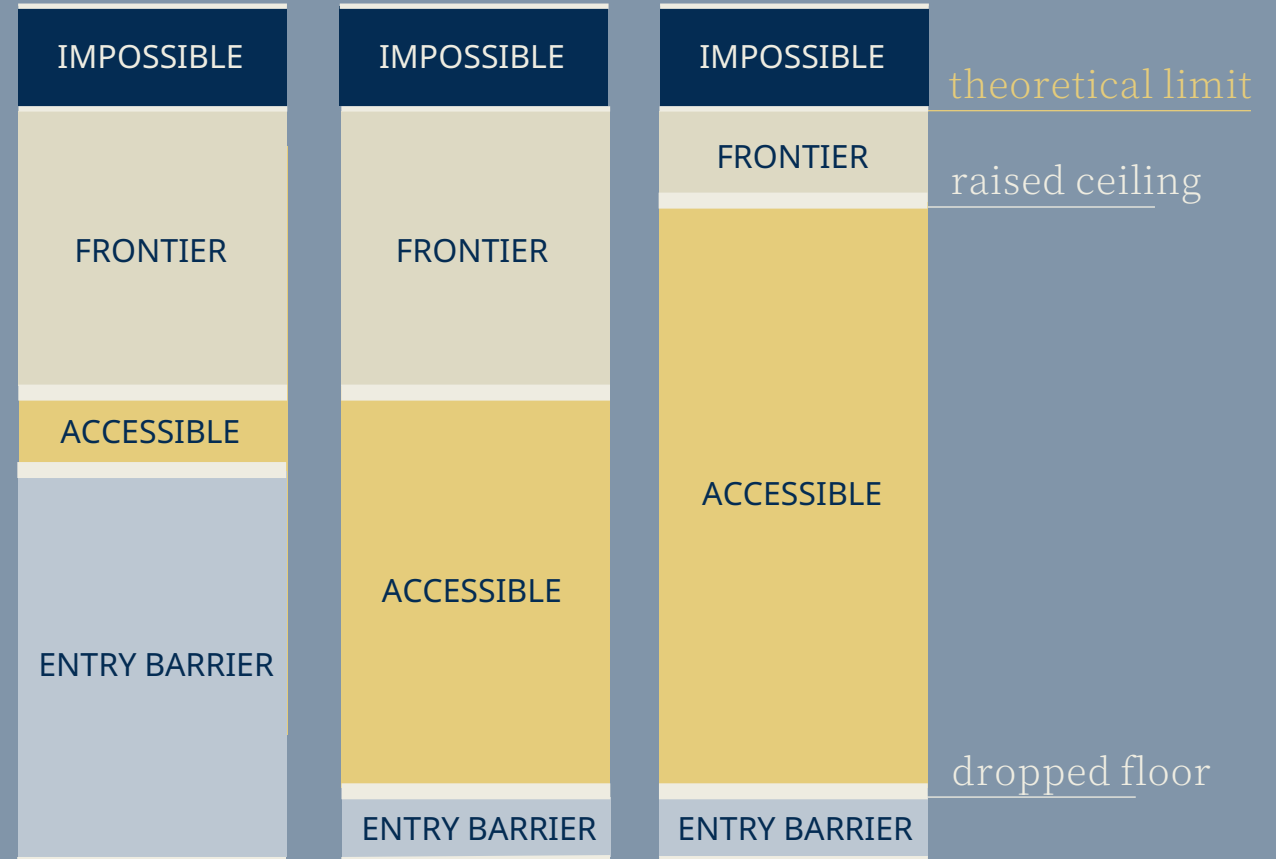
Agentic Guardrails

- Provable agentic **authorization**, **tool-use**, context-management, **execution safety** with:
 - privilege escalation
 - provenance & data-flow analysis
 - information-flow control
 - integrity & confidentiality
 - isolation
- Crisper **intent formalization**

When is privilege escalation
a feature and not a bug?

Takeaways

Yes. FM just got cheap



cheap $\neq$ easy

The floor has dropped

The ceiling is about to shoot up

The *theory's* the limit

harder problems

AI + FM will *reach* for the harder problems

More AI-generated code, more AI ops

More need for verification, safety, security

judgement

FM *reaches*, judgement *aims*

What to verify is the human call

Keep control where it matters

Next, I want to hear **your** views

claudiacauli@gmail.com

www.claudiacauli.com