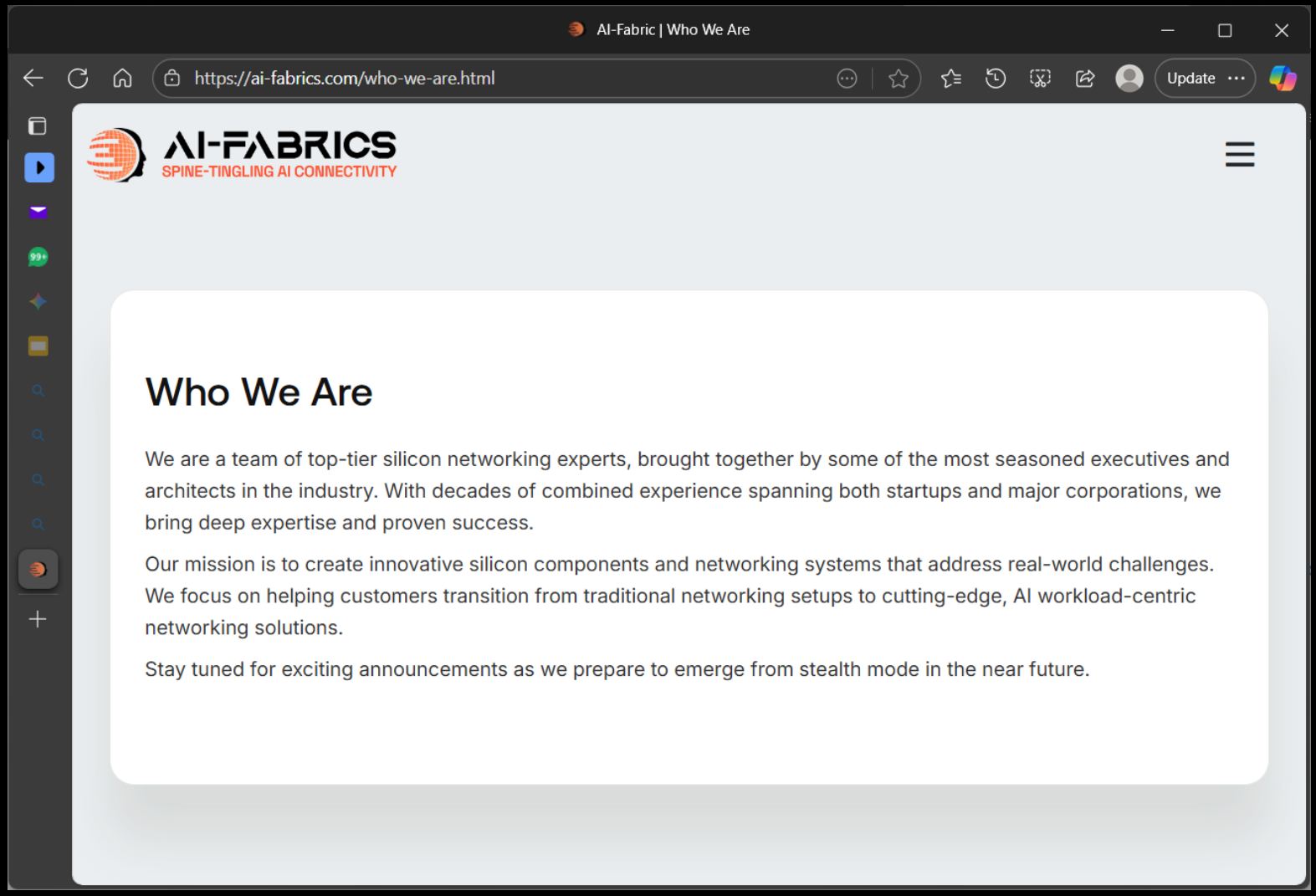
# DV Has a Playbook - Why Doesn’t FV?

Roy Frank
Verification Technical Lead
AI-Fabrics

SemIsrael Expo
Nov 11, 2025

AI-FABRICS
SPINE-TINGLING AI CONNECTIVITY


AI-Fabric | Who We Are
https://ai-fabrics.com/who-we-are.html
Update
AI-FABRICS
SPINE-TINGLING AI CONNECTIVITY
Who We Are
We are a team of top-tier silicon networking experts, brought together by some of the most seasoned executives and architects in the industry. With decades of combined experience spanning both startups and major corporations, we bring deep expertise and proven success.
Our mission is to create innovative silicon components and networking systems that address real-world challenges. We focus on helping customers transition from traditional networking setups to cutting-edge, AI workload-centric networking solutions.
Stay tuned for exciting announcements as we prepare to emerge from stealth mode in the near future.

# DV Has a Playbook - Why Doesn’t FV?

- DV has a clear methodology
- Formal doesn’t - but it could

- Design Verification (DV) has evolved into a disciplined practice
- Formal Verification (FV) is powerful but often an individual craft
- This talk explores the benefits of a structured methodology - and general ideas of what a formal ‘playbook’ could look like

# In This Talk

- Two Worlds of Verification
- DV: Why UVM Succeeded
- FV: Formal Verification Today
- What's Missing: The Playbook Gap
- The DV Playbook
- Building a Formal Playbook
- Conclusion

# Two Worlds of Verification

- DV and FV share goals but differ in structure

- DV: Mature, process-driven, measurable
- FV: Powerful, precise, but unpredictable
- → Why does one have a playbook and the other doesn’t?

# Why UVM Succeeded - Lessons for Formal

- **UVM turned verification from coding into a disciplined engineering practice.**

- **Key success factors:**
  - Standardization: unified methodology across vendors (Accellera)
  - Modularity: reusable components (agents, monitors, scoreboards)
  - Common language across teams
  - Metrics: coverage-driven closure, definition of “done”
  - Automation: regressions, random stimulus, factory overrides
  - Process Discipline: hierarchy, reviews, documentation
  - Community & Ecosystem: shared training and adoption

- **Lesson for Formal:**
  → Build structure, reuse, and measurable progress - not just powerful tools

# Formal Verification Today

- Formal is still too dependent on individual expertise.

- Strengths: exhaustiveness, precision, bug depth
- Weaknesses: narrow adoption, scalability
- Seen as ‘magic’ — unpredictable at scale

# What's Missing: The Playbook Gap

- FV lacks structure and shared language
- Reinvented process in every
  - Company
  - Project/team
  - Team members
- Not enough reuse
- Need to scale up
  - From signal level to transaction level

    assert ( pkt_ended |-> tx_pkt.data == ref_pkt.data );

    When the underlying protocol changes, update the agent and no need to modify all my checkers
- Community
- This calls for 'FVM' – a formal playbook
  - the equivalent of UVM

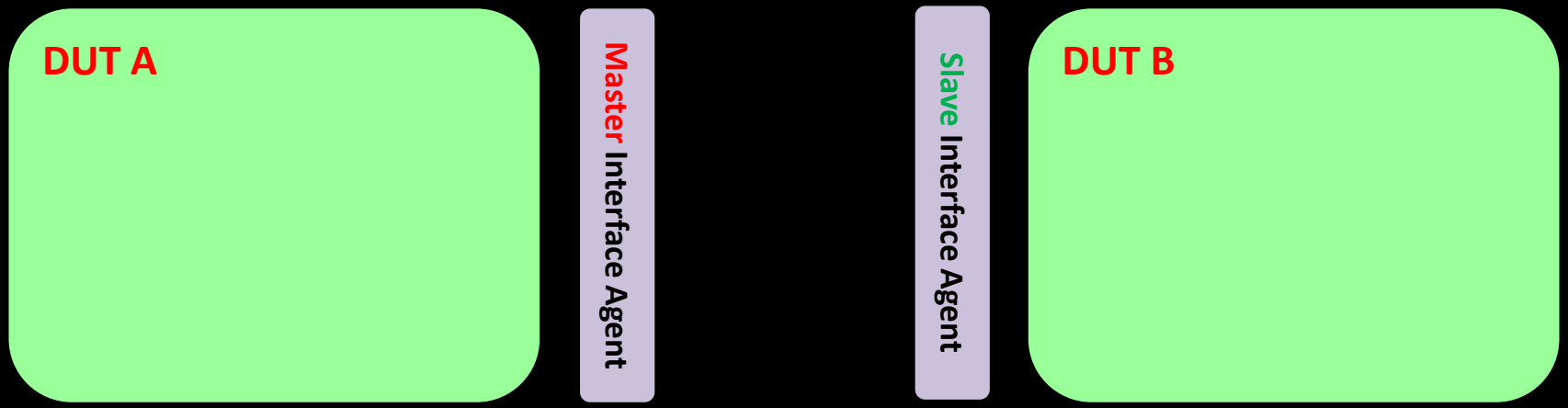
# Formal Playbook Framework

- Enable common infrastructure and reuse
- Introduce standard components
  - Assertion libraries
  - Property templates
  - RAL
  - Formal Scoreboards
  - Agents, checkers, ref models etc.

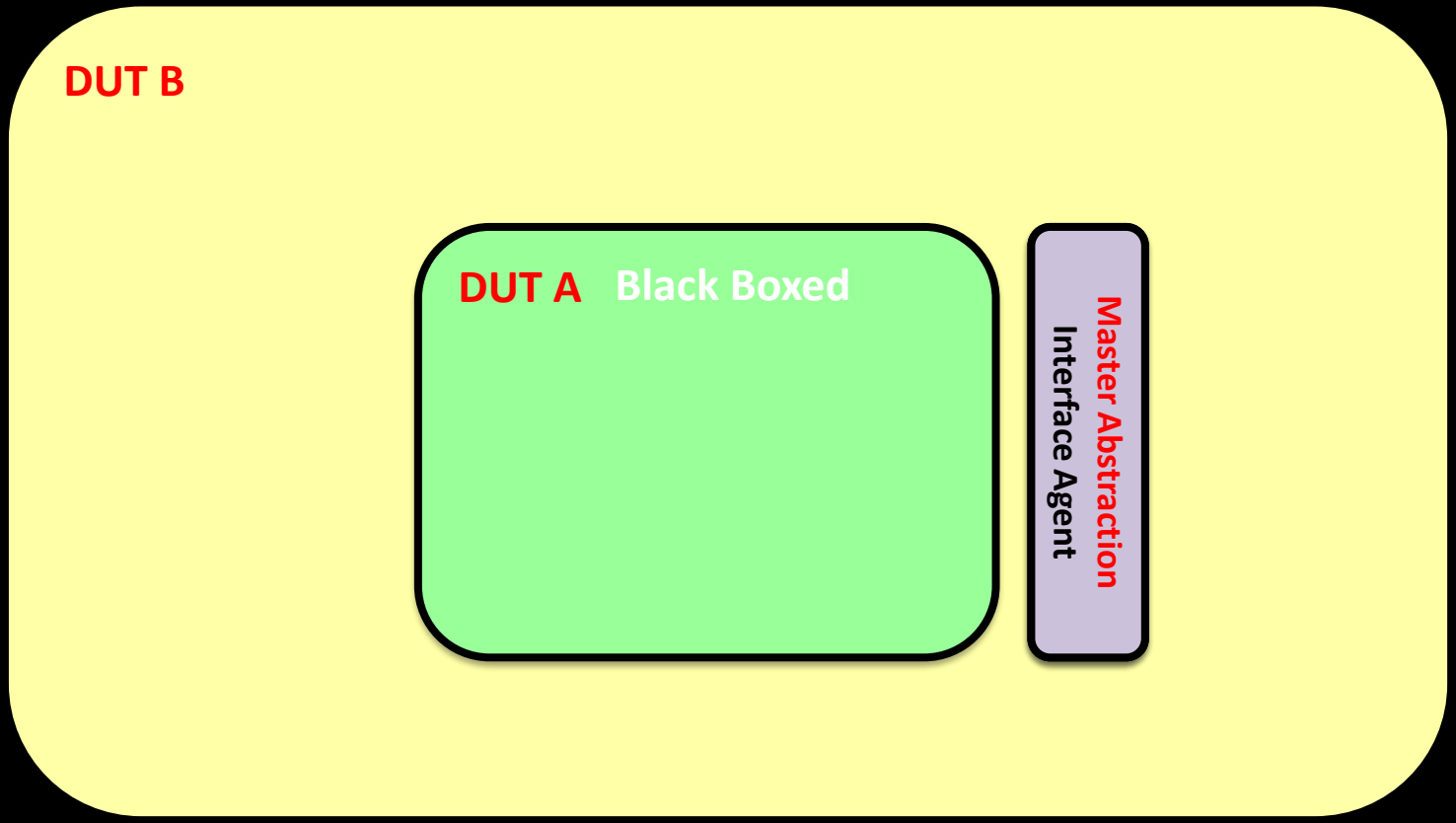
# Formal Playbook Framework - cont.

- Scale up
  - From local property scope to components
    - Similar to UVM's agents for interfaces
  - Need clear and well-defined API between components
    - Agents to checkers, reference models etc.
- Reuse would make formal more scalable
- Turn one-off proofs into reusable modules
- Reusability + automatic Assume Guarantee cycles
- No need to re-invent the wheel

# Reuse 1 – Master Slave Switch

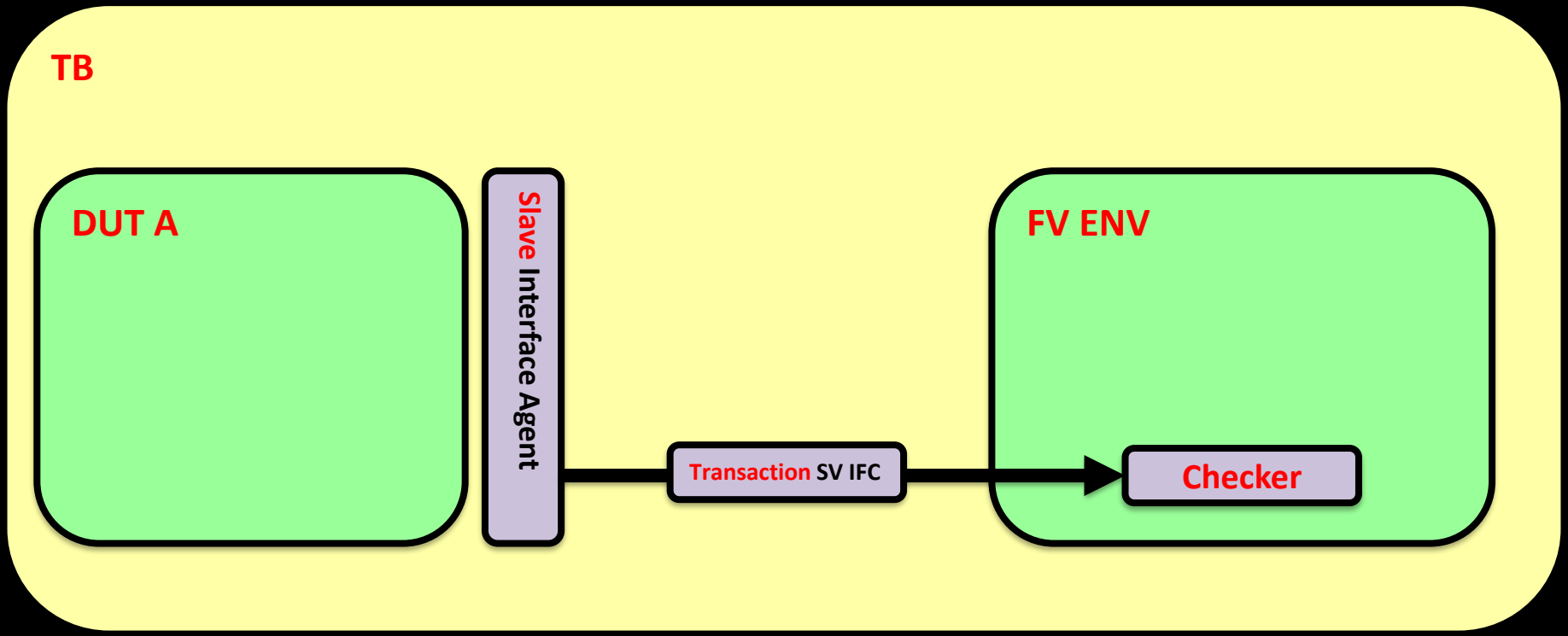


# Reuse 2 – Abstraction Model



# Reuse 3 – Interface Transaction

**No more sticking to signal level**



# Formal Verification Community

- There are numerous DV communities
- Fewer ones in FV
  - It's about scale
  - But not only

# Formal Verification Community

- Community + a common methodology (UVM like?) would:
  - Create a **common language**
  - **Share** formal building blocks between teams/companies
  - Reduce **learning curves** for new employees
  - Boost formal **productivity**

# Conclusion: From Ad Hoc to Methodology

- FV should achieve DV-level maturity through structure

- DV matured via process — FV can too
- Build consistency, share successes
- Make formal a first-class citizen in verification

# THANK YOU!

AI-FABRICS
SPINE-TINGLING AI CONNECTIVITY