



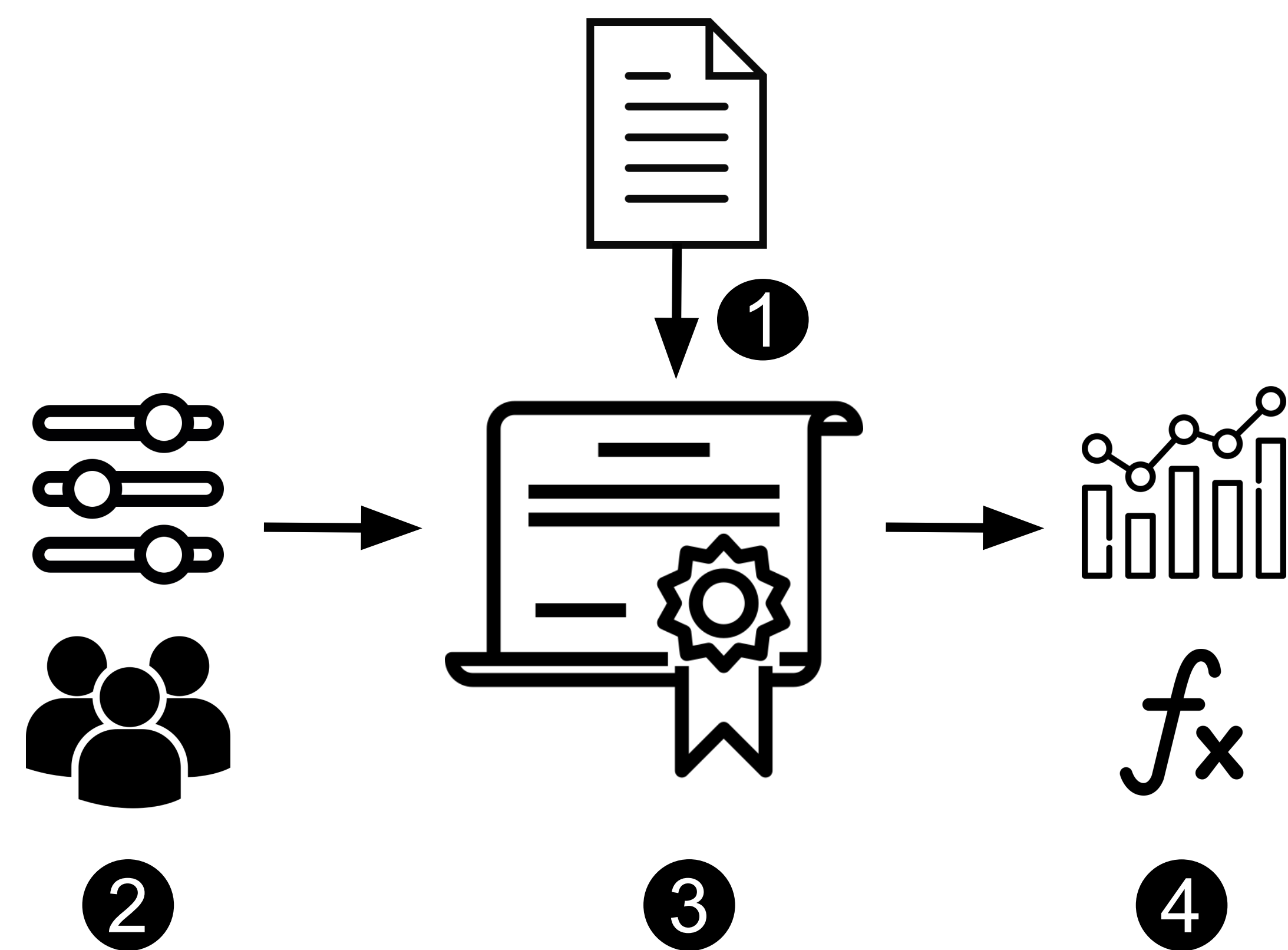
Rishabh Iyer, Luis Pedrosa, Arseniy Zaostrovnykh, Solal Pirelli, Katerina Argyraki, George Candea

Context

- ❖ HW networking: Consistent performance but no programmability
- ❖ SW networking: Flexible but unexpected performance behaviour

Goal

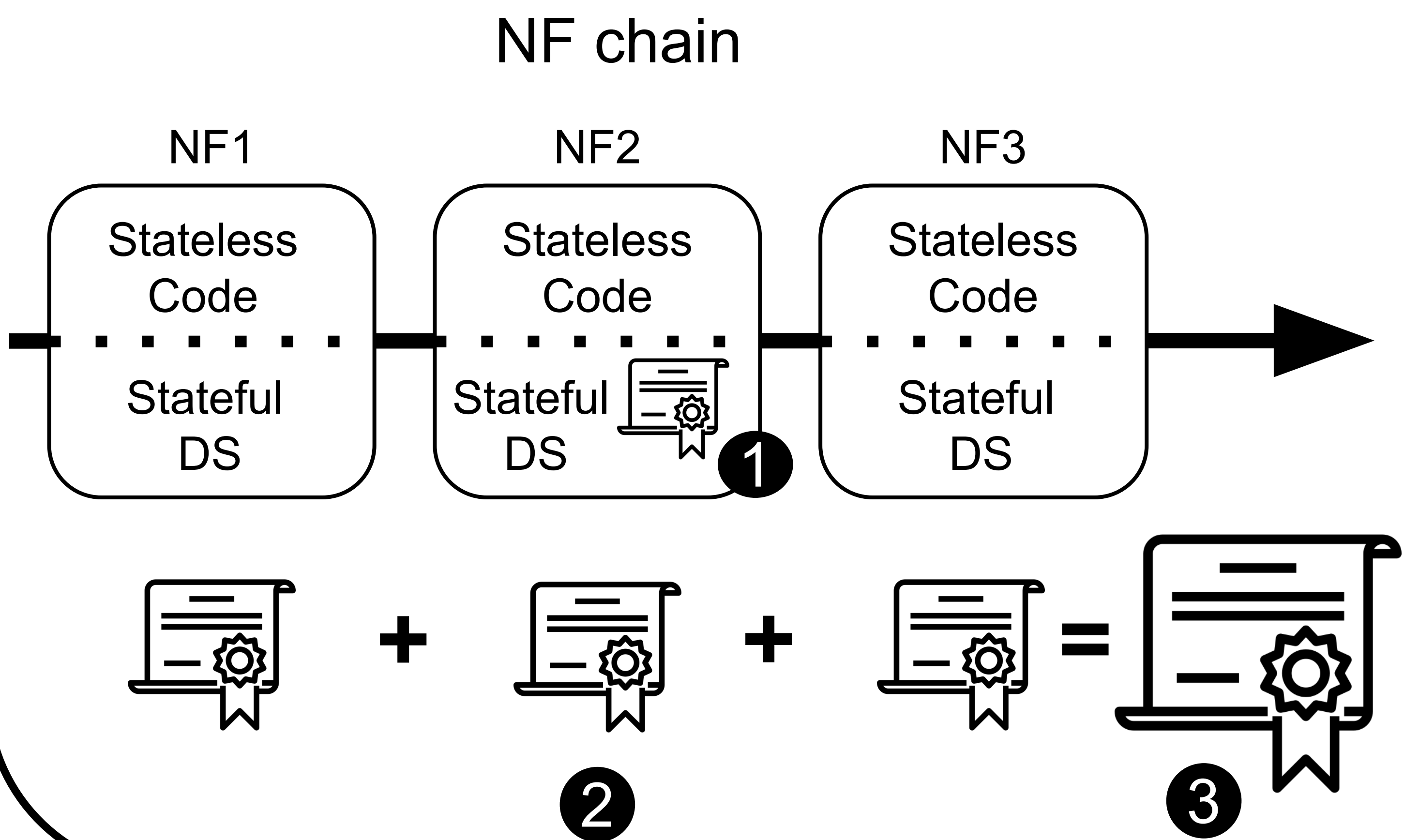
- ❖ Comprehensive understanding of entire performance profile of NF



Performance Contracts - Workflow

1. Generate contracts from NF code
2. Users parameterize arbitrary input workloads
3. Contracts predict perf for workload w/o running NF
4. Performance predicted as function of **Performance Critical Variables** (PCVs)

Generating Performance Contracts - A Recursive Process

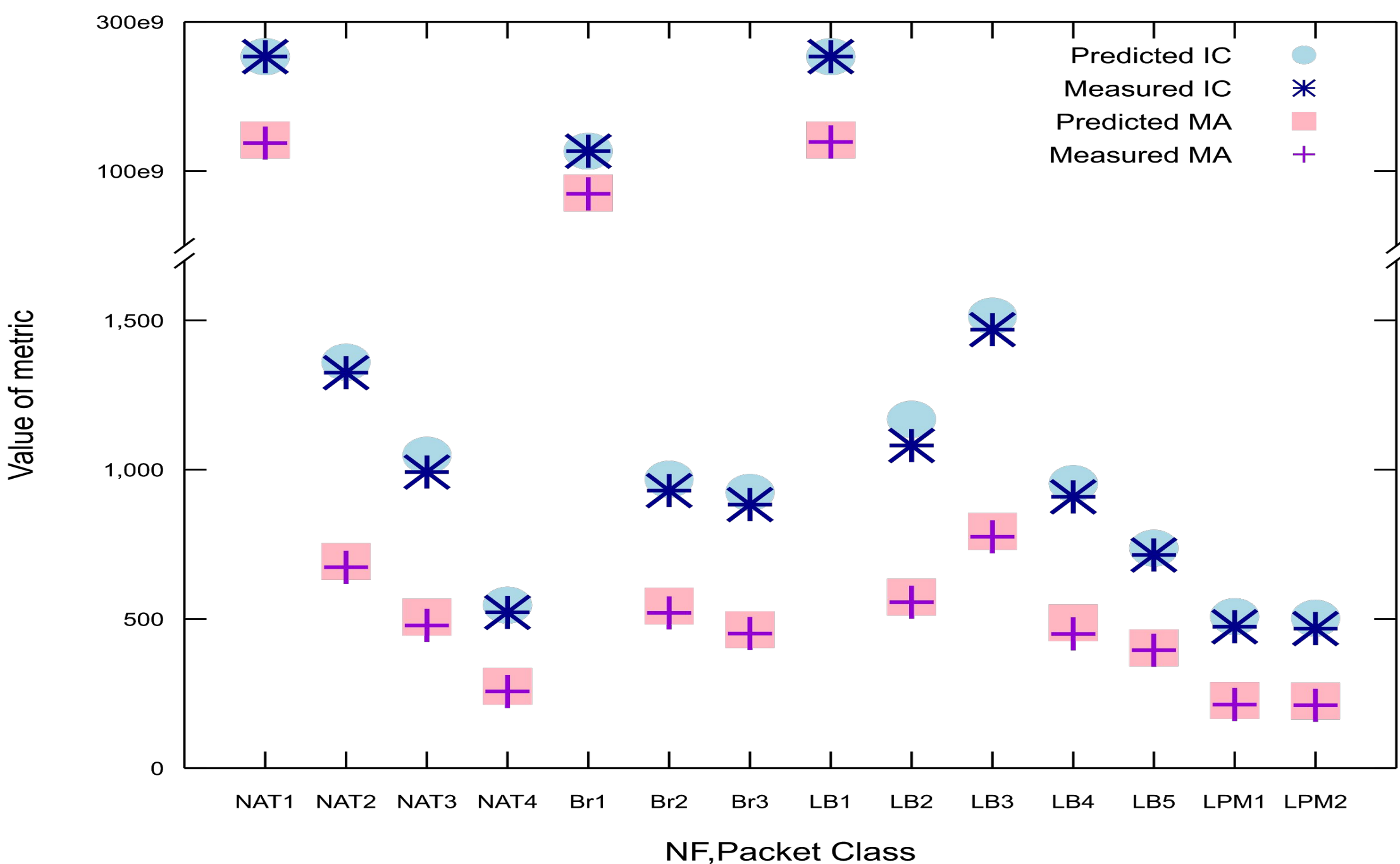


1. **Base Case: Stateful Data structures**
 - i. PCV abstraction avoids path explosion
 - ii. Distill key factors that impact performance
2. **Contracts for NFs**
 - i. Exhaustively Symbex stateless code; tally cost
 - ii. Plug-in contracts for stateful functions
3. **Contracts for NF chains**
 - i. Pair packet classes from connected NFs
 - ii. Packet sent (1st NF) == Packet received (2nd NF)

Evaluation

4 NFs: NAT, Maglev-like LB, Bridge, LPM Router
Metrics: Instruction Count, Memory Accesses, Latency

Prediction Accuracy for Instr Count (IC), Mem Accesses (MA)



Max prediction gap - 7.5% (IC) and 7.6% (MA)

Use Case Scenarios

Better tools for Network Operators and NF Developers!

Network Operators:

1. Visualize NF performance under attack
2. Visualize performance of NF chain configuration

NF Developers:

1. Debug workload-specific perf bottlenecks
2. Pick appropriate DS for expected workload