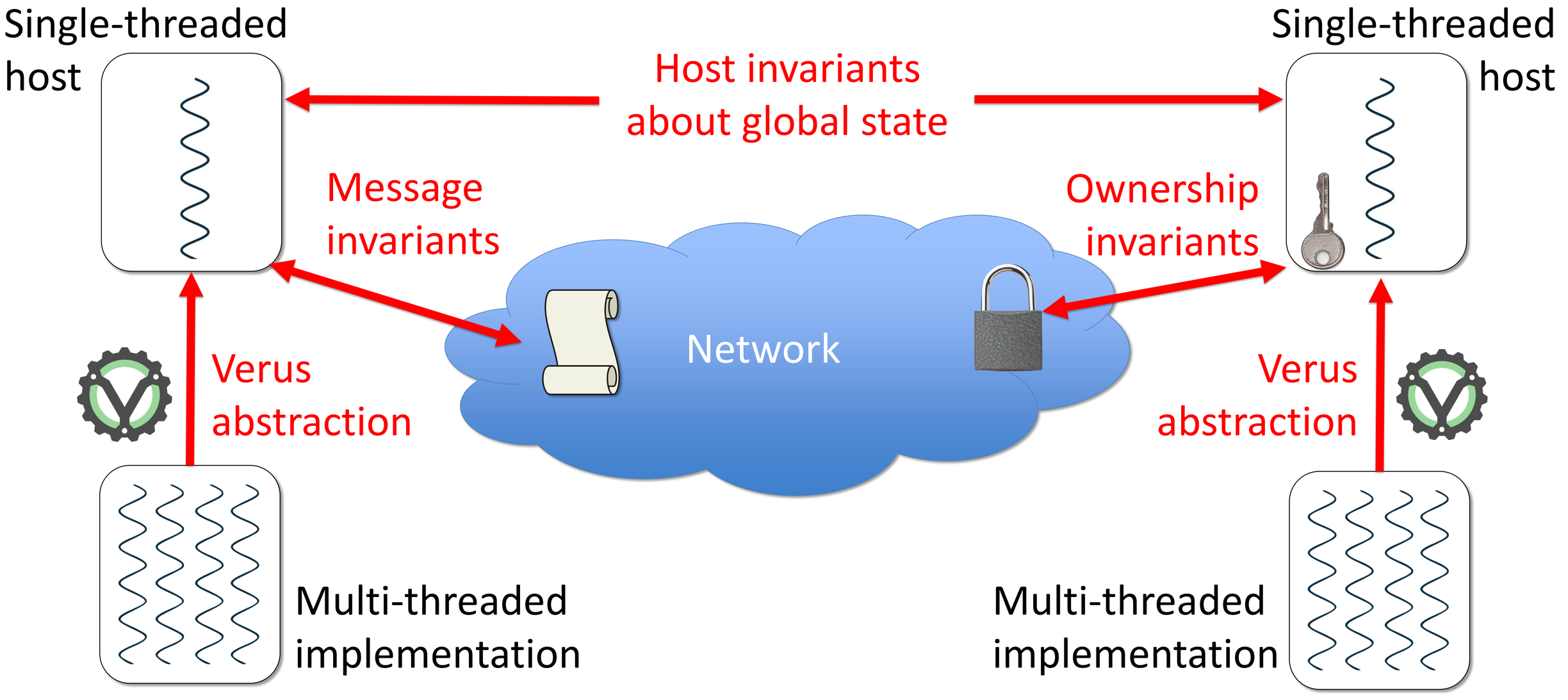
Simplifying End-to-End Verification of High-Performance Distributed Systems

Bryan Parno (CMU), Manos Kapritsos (University of Michigan)



Challenge

Distributed systems are hard to reason about, due to:

- Asynchronous, unreliable networks
- State is distributed across multiple machines
- Complexities of real-world, multi-threaded implementations

Developers need *copious manual effort and expertise* to prove such systems correct: a **major obstacle to the adoption of formal verification**

Scientific impact

- Identified an *invariant taxonomy* that allows for faster identification of inductive invariants [OSDI'24]
- Developed **Verus**, an automated tool for *verifying (concurrent) systems software*
  - Handles safe and unsafe Rust
  - Reduces developer burden
  - Produces verification results orders of magnitude faster [SOSP'24]
  - Automated debugging for failed proofs [CAV'24, Distinguished Paper Award]

Two of three *Best Paper awards* at OSDI'24 used Verus

Societal impact

- Simpler protocol verification means more reliable digital infrastructure
- Making verification easier brings us closer to a future where mission-critical software is verified, not merely tested

Education and outreach

- Developed *a summer school and a class* on formal verification of systems
- **WIP**: *A textbook* on formal verification of systems
- **WIP**: A day-long tutorial on Verus

Broader participation

Verus makes systems verification available to a broad community of developers

Since it is based on Rust, Verus makes the "jump" to formal verification much easier for developers