Pattern-based Abstractions for Parameterized Model Checking of Distributed Algorithms

Thanh Hai Tran, Jure Kukovec

TLA+

- Underlying theories
  - Set theory (ZFC)
  - Linier-time logic

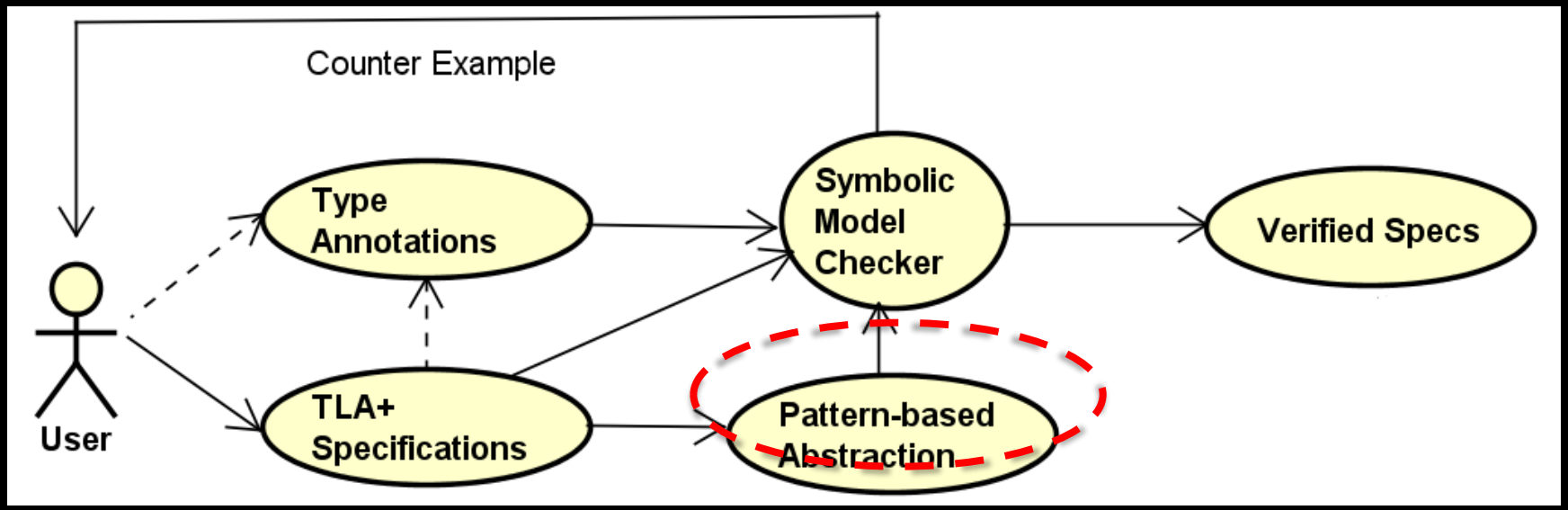
$$
\begin{array}{l}
Init \triangleq \\
\quad \land\ sent = \{\} \\
\quad \land\ pc \in [Proc \to \{\text{“V0”}, \text{“V1”}\}] \\
\quad \land\ rcvd = [i \in Proc \mapsto \{\}]
\end{array}
$$

$$
\begin{array}{l}
Receive(self) \triangleq \\
\quad \land\ newMsgs' \in \text{SUBSET}\ (sent \cup ByzMsgs) \\
\quad \land\ rcvd' = [i \in Proc \mapsto \text{IF}\ i \neq self\ \text{THEN}\ rcvd[i] \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE}\ \ rcvd[self] \cup newMsgs']
\end{array}
$$

- Tools: TLC (model checker), TLAPS (theorem prover)
- Industrial projects: Paxos, Alpha EV7/EV8…

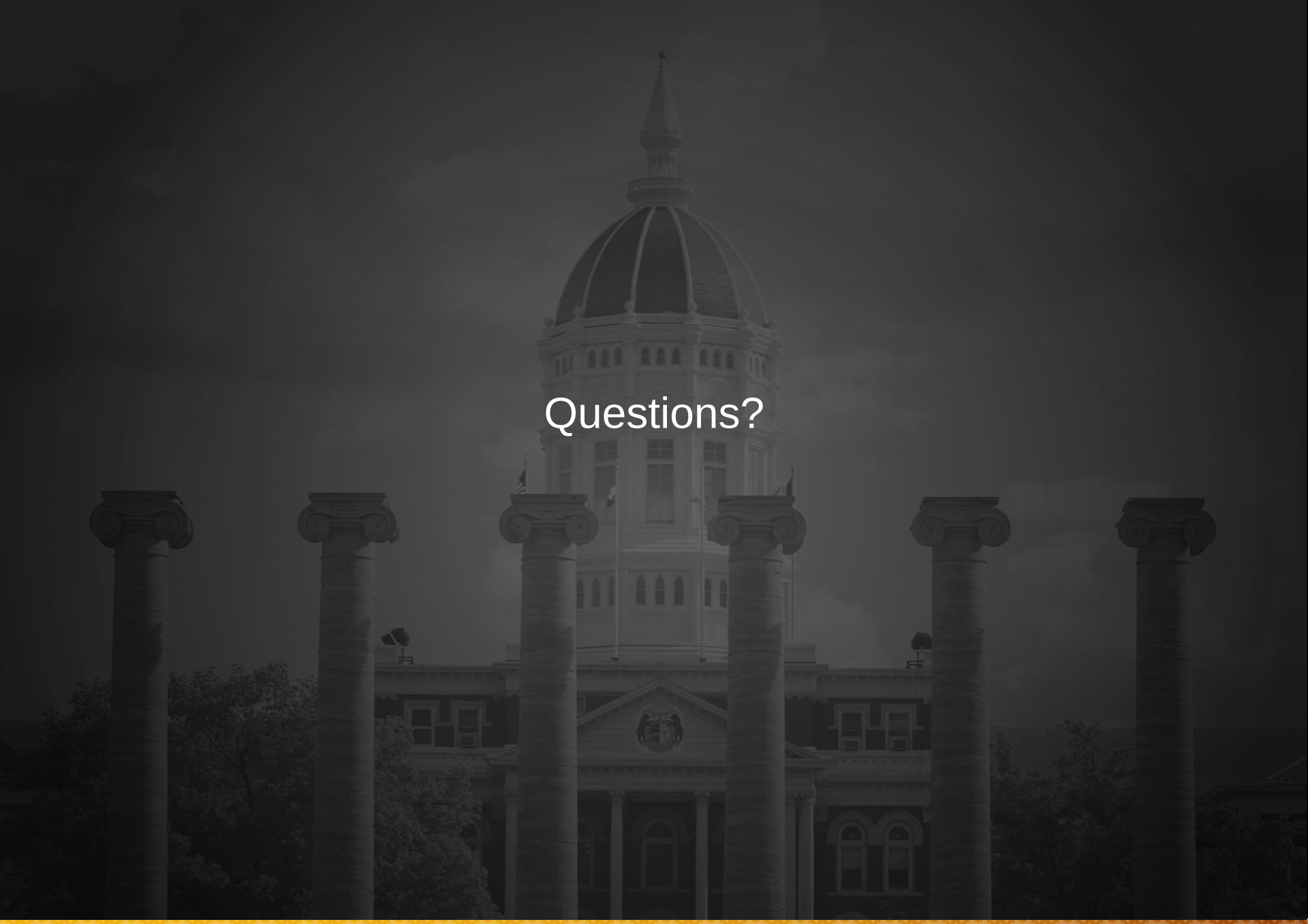
Goals: Pattern-based Abstractions

- TLA+ patterns
- Pattern-based abstractions
- Fault-tolerant distributed algorithms



Challeges

- TLA+ features: sequences, set cardinality, CHOOSE...
- Type systems
- TLA+ patterns
- Industrial-strength algorithms


Questions?