Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes
research

Predicting Performance Degradations of Black-Box Microservice Applications

11th Symposium on Software Performance 2020

Martin Straesser, Johannes Grohmann

13 November 2020

Leipzig, Germany

https://se.informatik.uni-wuerzburg.de

Problem: Reactive Monitoring of Microservices

- Microservice applications as main architectural paradigm for cloud applications [1]

PINPOINT

[2]

- Performance engineers rely on **reactive** application performance management (APM) tools

[3]

- **Proactive** performance management needed to ensure user experience and revenue

[4]

[1] S. Eismann et al.: Microservices: A Performance Tester's Dream or Nightmare? ICPE'20, 2020
[2] Pinpoint APM by Naver Corp. Picture Source: https://blog.naver.com/varkiry05/221441388036
[3] Kieker Monitoring by W. Hasselbring and A. van Hoorn. http://kieker-monitoring.net, Picture Source: https://twitter.com/kiekerapm
[4] Zipkin Distributed Tracing System. Picture Source: https://zipkin.io/

SuanMing: Enabling Proactiveness for APMs

- *SuanMing* augments an existing reactive monitoring stack with a **proactive component**
- Weaknesses of related work (e.g. [5-10])
  - Lack of explainability
  - Require low-level hardware measurements or application logs

- Goals
  - High **explainability**
  - **No prior knowledge**
  - **Modular** and **extensible**

- Prediction process divided into four steps

[5] A. Jindal et al.: Performance Modeling of Cloud Microservice Applications. ICPE'19, 2019
[6] J. Lin et al.: Microscope: Pinpoint Performance Issues with Causal Graphs in Microservice Environments. Service-Oriented Computing, 2018
[7] Y. Gan et al.: Seer: Leveraging Big Data to Navigate the Increasing Complexity of Cloud Debugging. ASPLOS'19, 2019
[8] T. Pitakrat et al.: Hora: Architecture-aware online failure prediction. Journal of Systems and Software, 2018
[9] H. Jayathilaka et al.: Performance Monitoring and Root Cause Analysis for Cloud-hosted Web Applications. WWW'17, 2017
[10] L. Wu et al.: MicroRCA: Root Cause Localization of Performance Issues in Microservices. NOMS'20, 2020

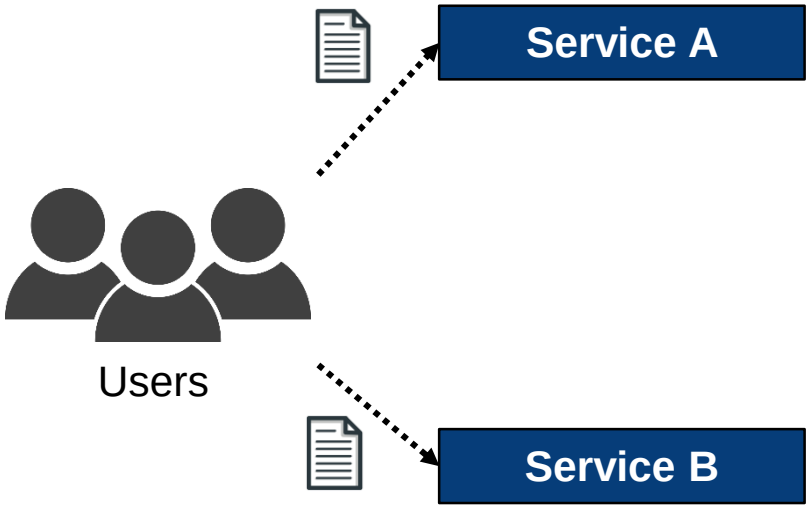
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › *Root-Cause Detection*

Users

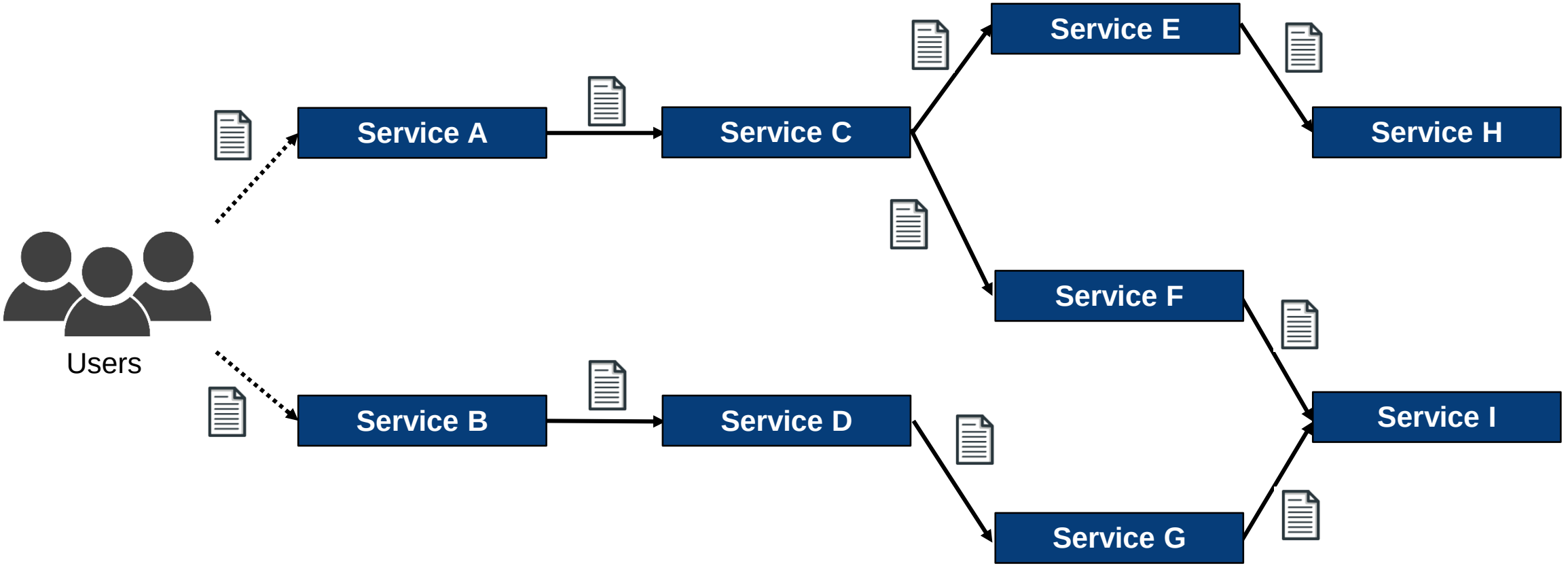
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › *Root-Cause Detection*



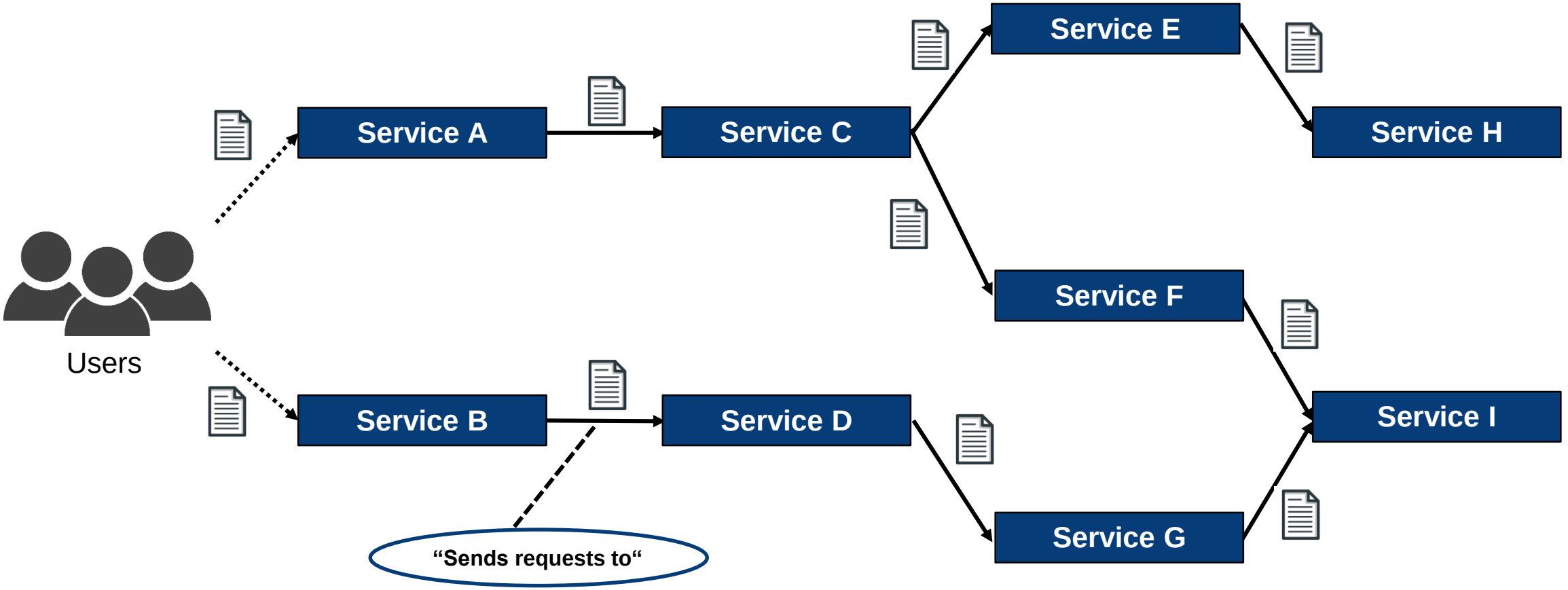
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › *Root-Cause Detection*



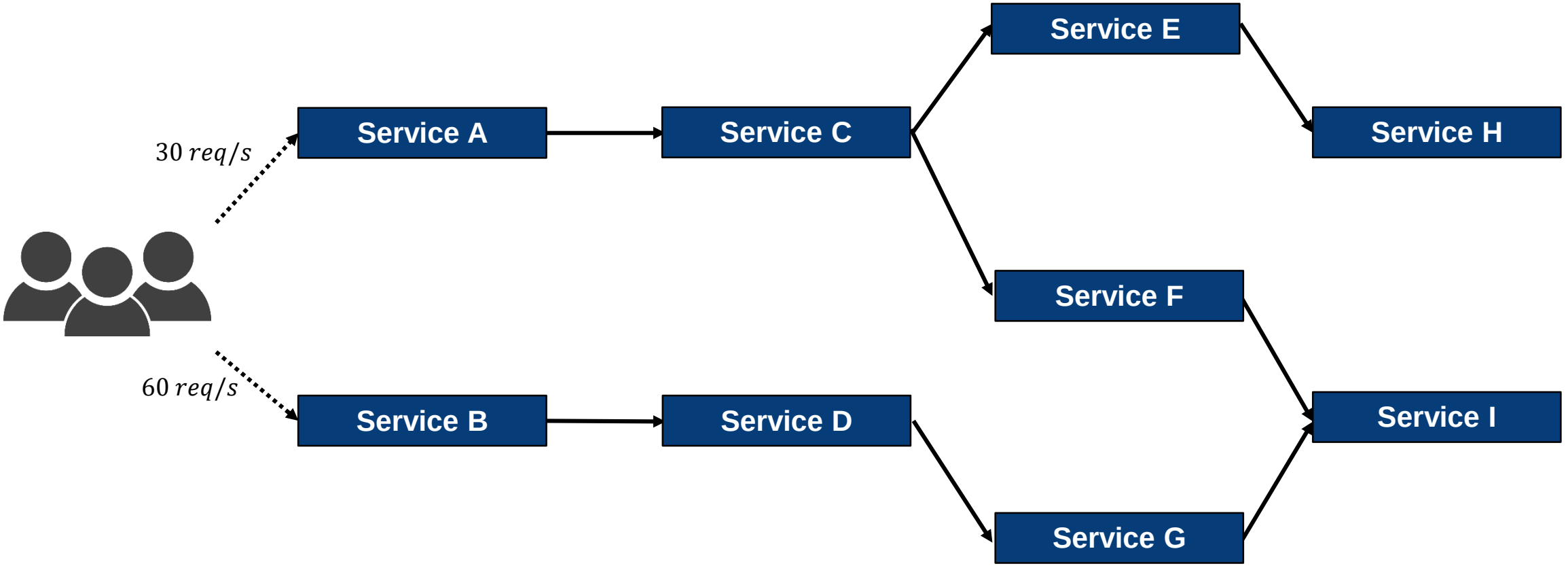
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › *Root-Cause Detection*



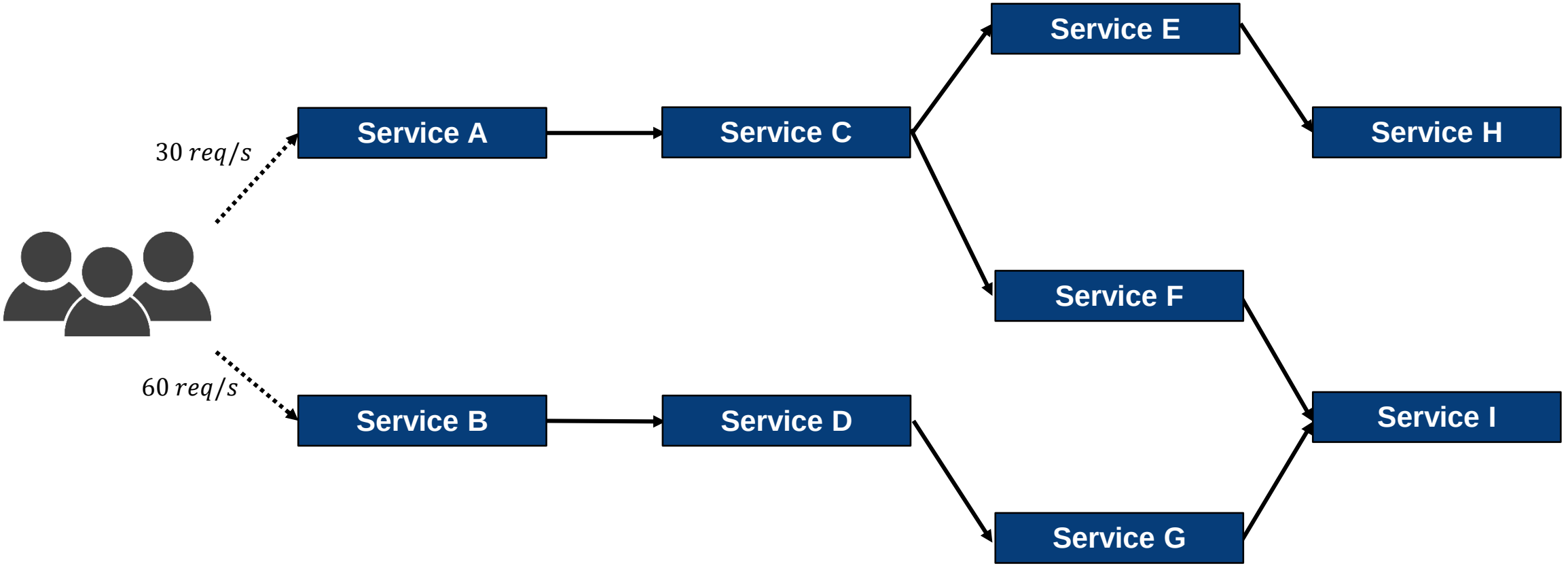
SuanMing: Predicting Performance Degradations

Initialization › **Load Forecasting** › *Request Propagation* › *Performance Inference* › *Root-Cause Detection*



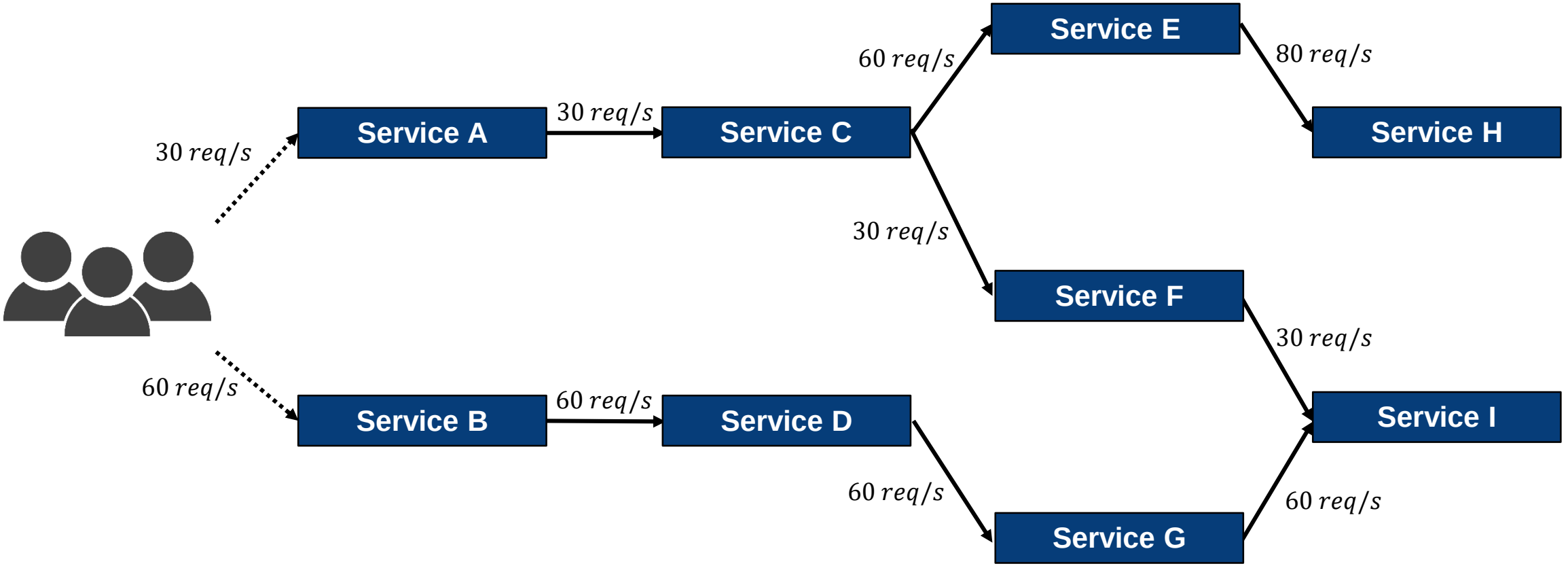
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › **Request Propagation** › *Performance Inference* › *Root-Cause Detection*



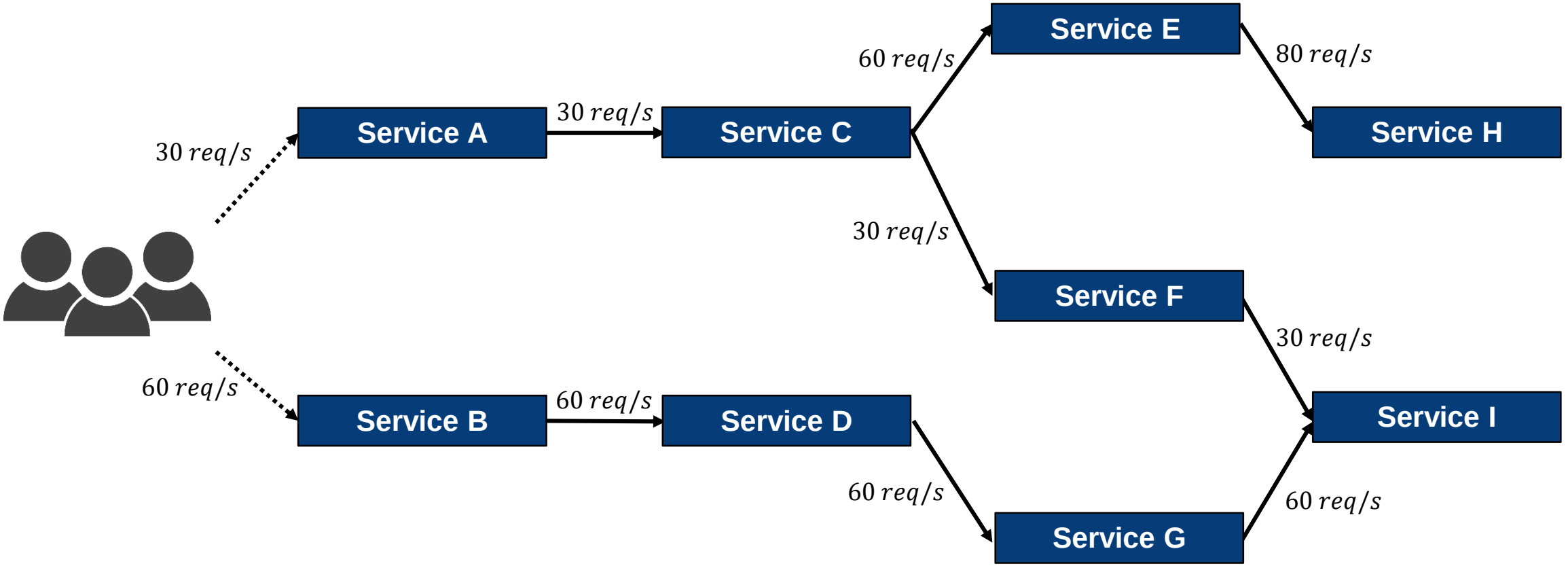
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › **Request Propagation** › *Performance Inference* › *Root-Cause Detection*



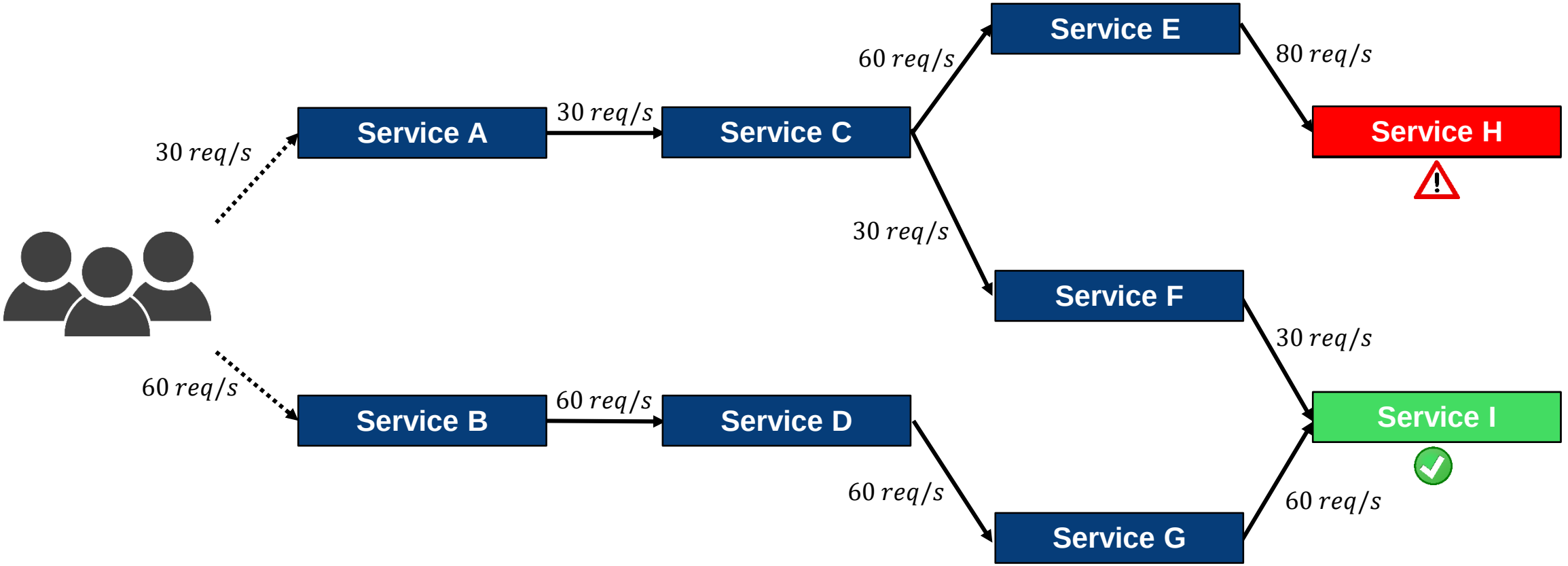
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › **Performance Inference** › *Root-Cause Detection*



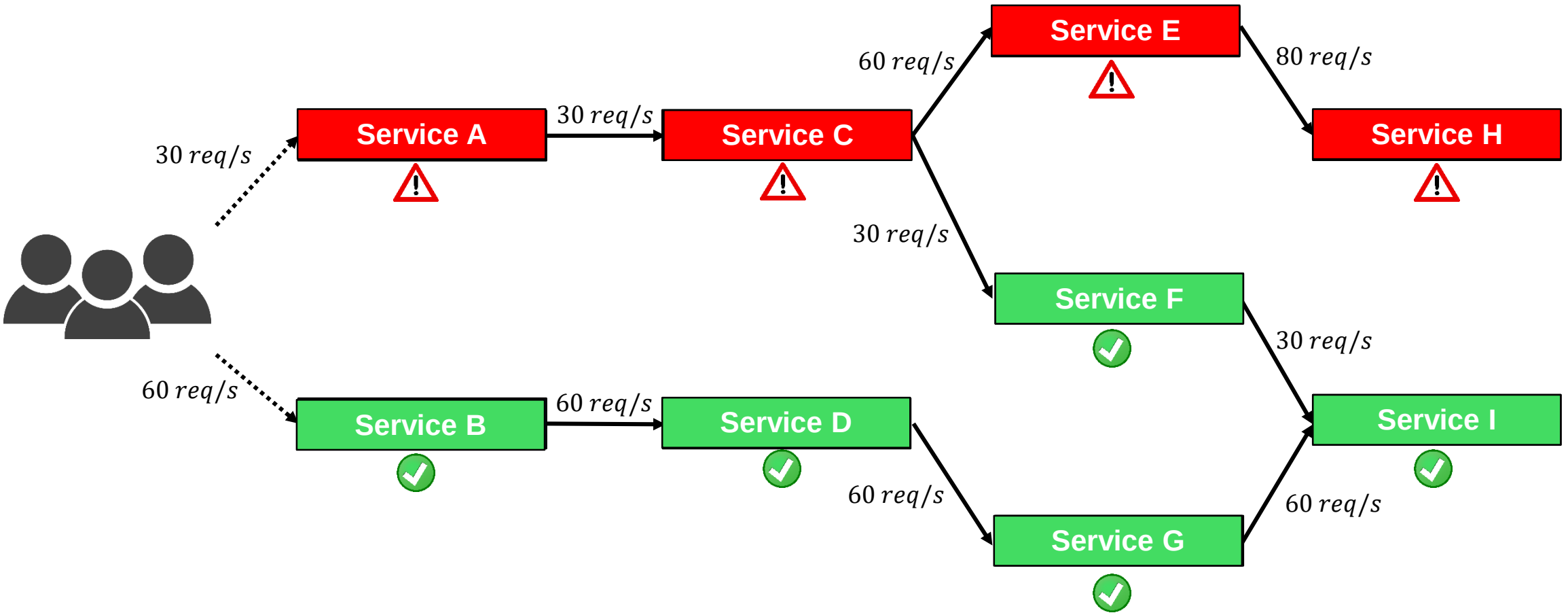
SuanMing: Predicting Performance Degradations

Initialization › Load Forecasting › Request Propagation › **Performance Inference** › Root-Cause Detection



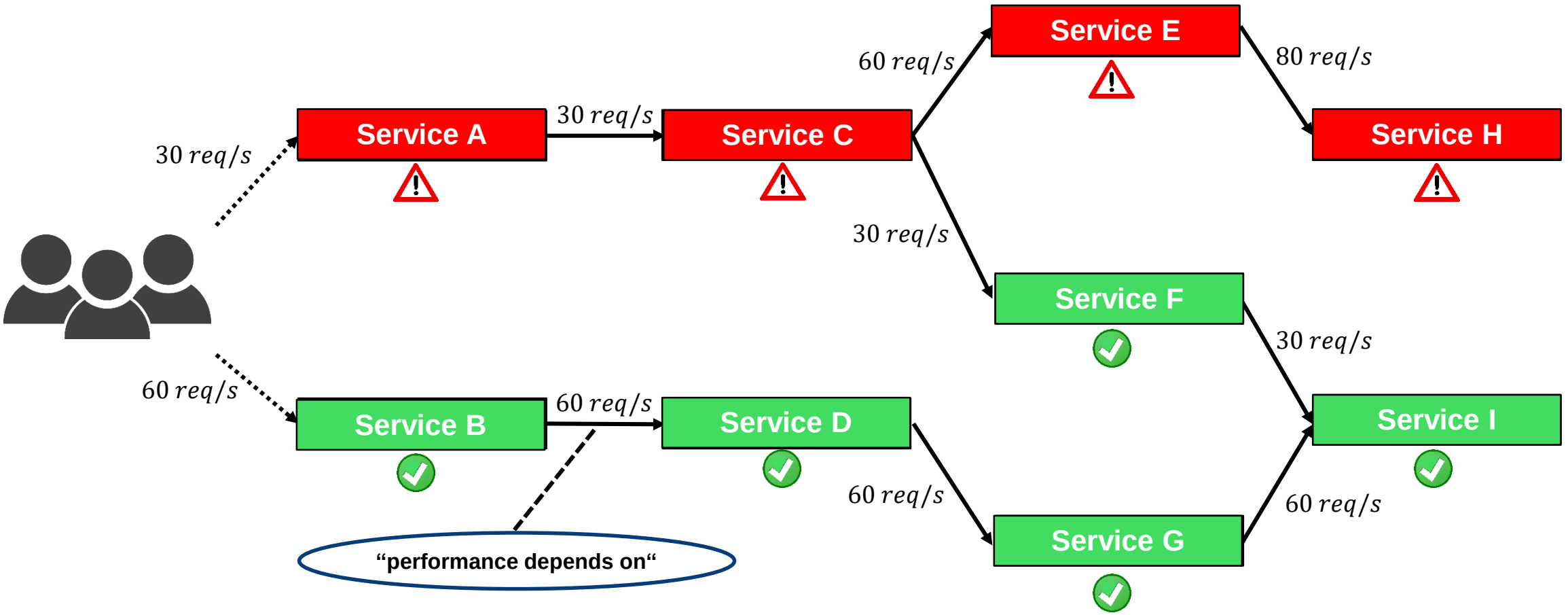
SuanMing: Predicting Performance Degradations

Initialization › Load Forecasting › Request Propagation › **Performance Inference** › Root-Cause Detection



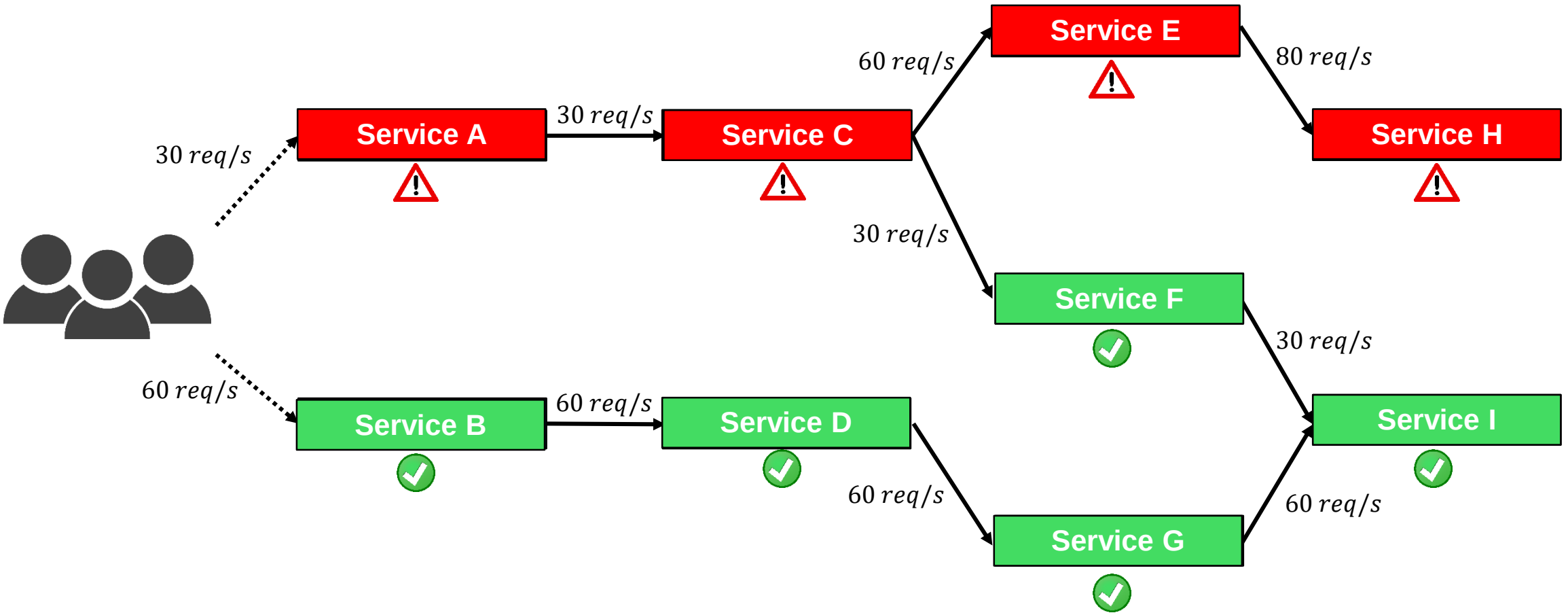
SuanMing: Predicting Performance Degradations

Initialization › Load Forecasting › Request Propagation › **Performance Inference** › Root-Cause Detection



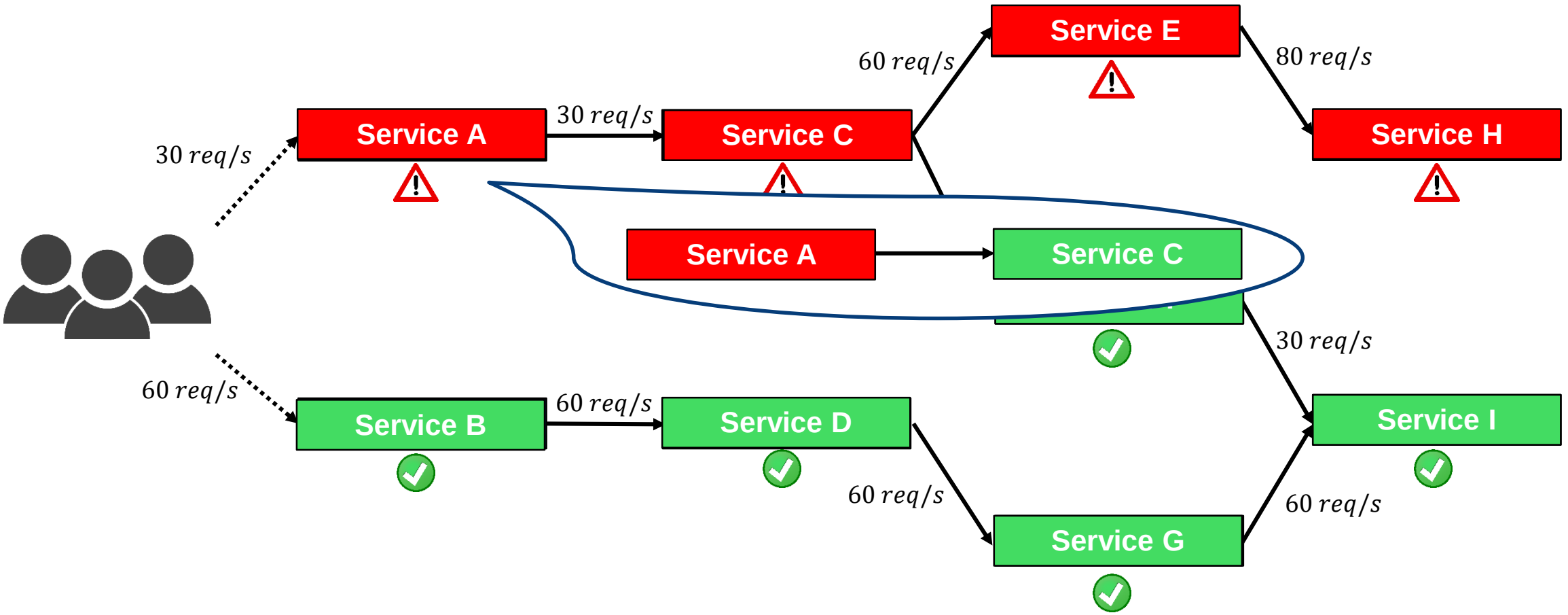
SuanMing: Predicting Performance Degradations

Initialization › Load Forecasting › Request Propagation › Performance Inference › **Root-Cause Detection**



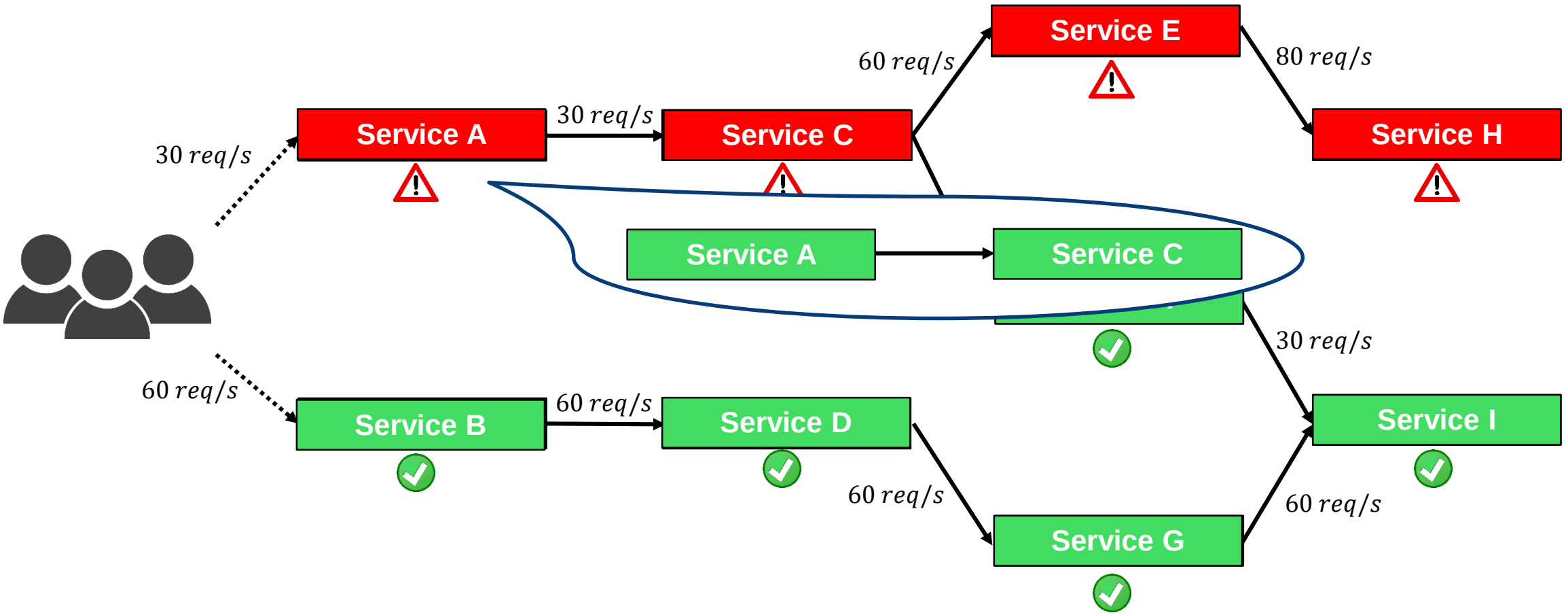
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › **Root-Cause Detection**



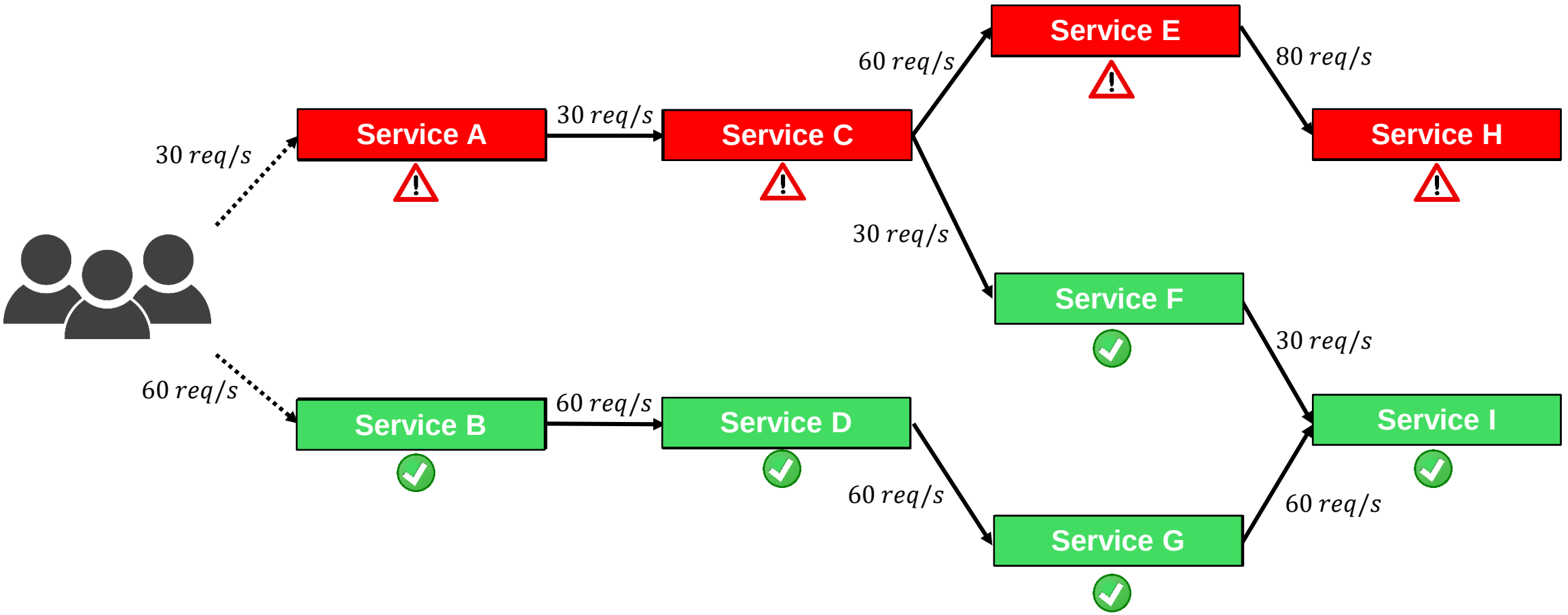
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › **Root-Cause Detection**



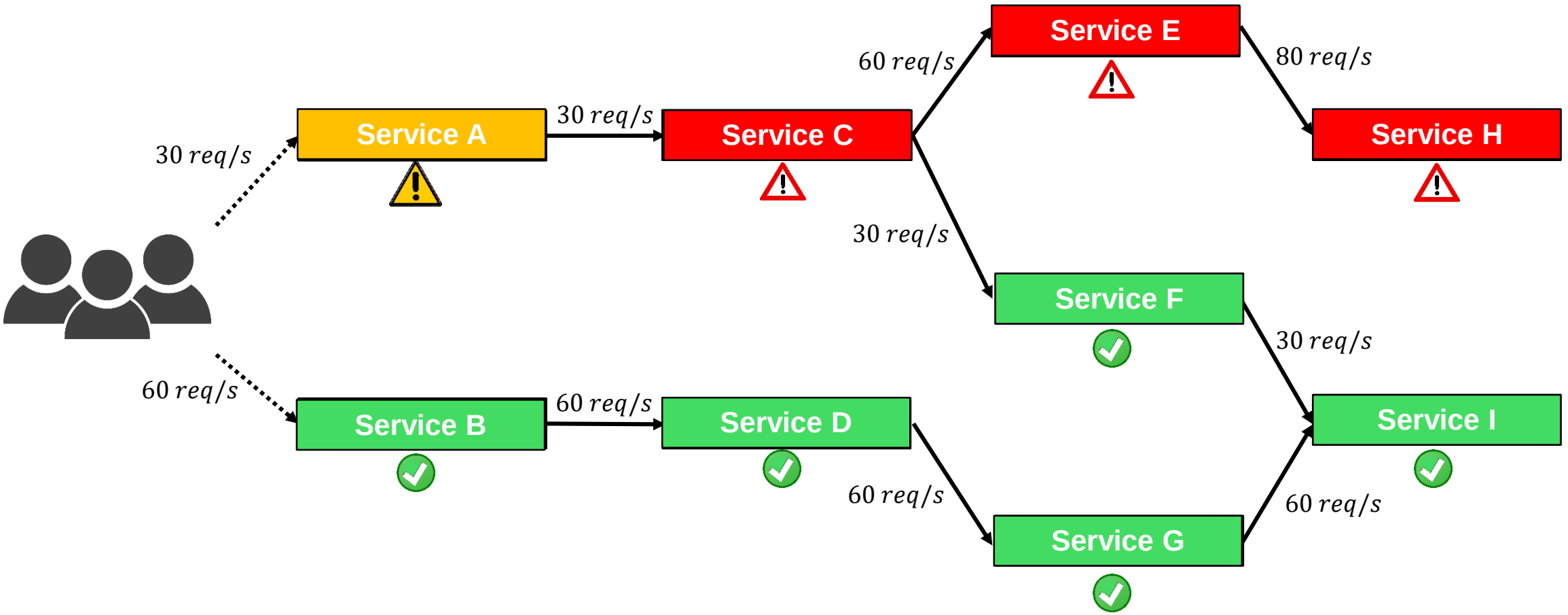
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › **Root-Cause Detection**



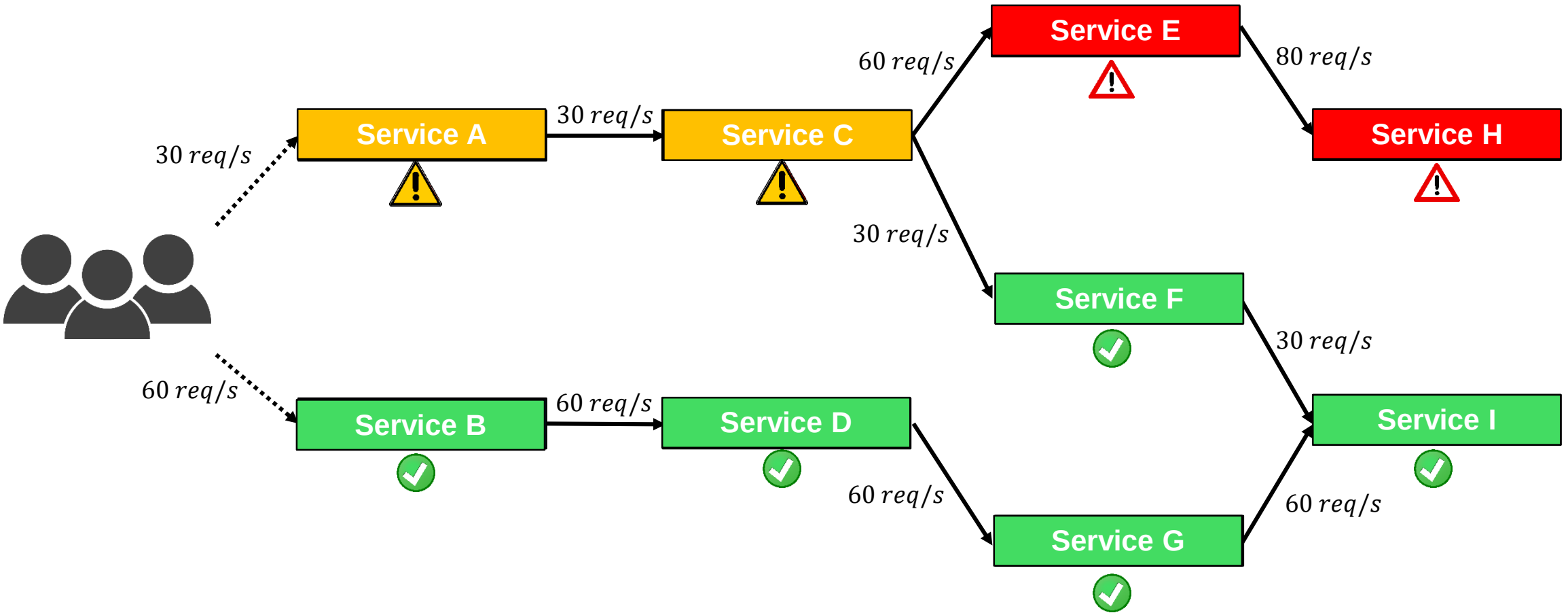
SuanMing: Predicting Performance Degradations

Initialization › Load Forecasting › Request Propagation › Performance Inference › **Root-Cause Detection**



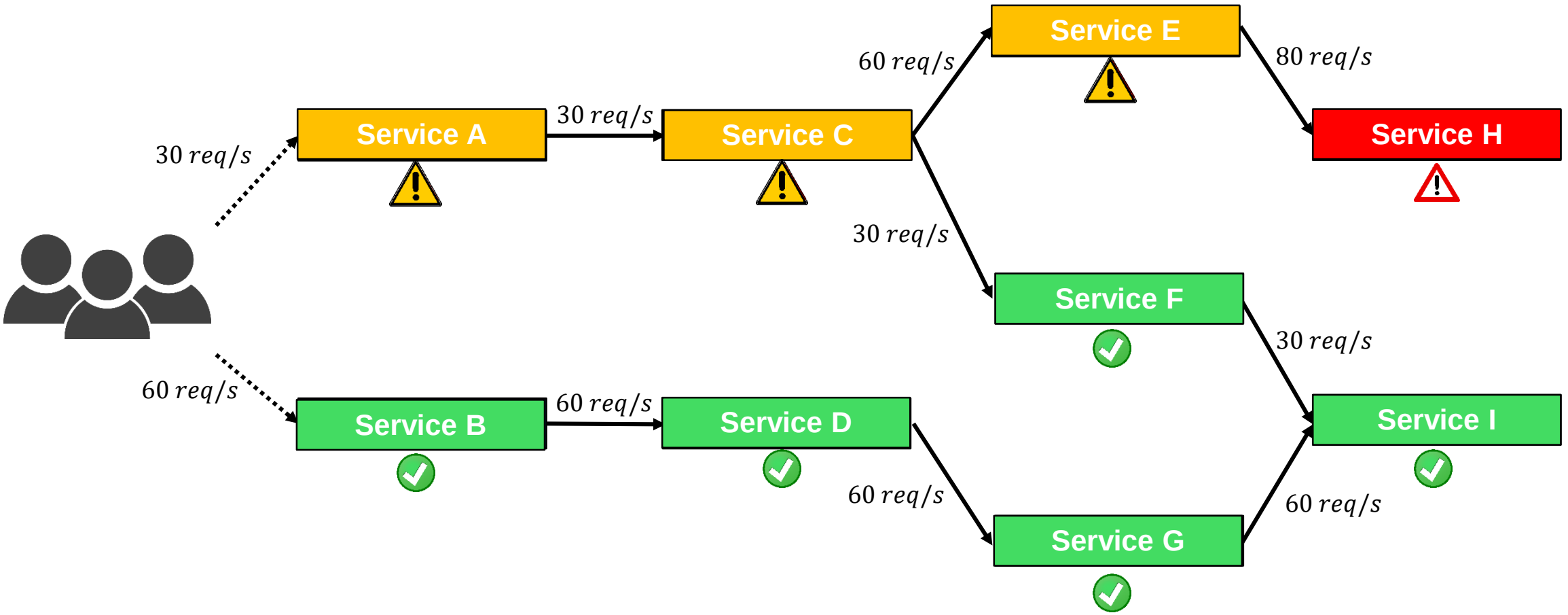
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › **Root-Cause Detection**



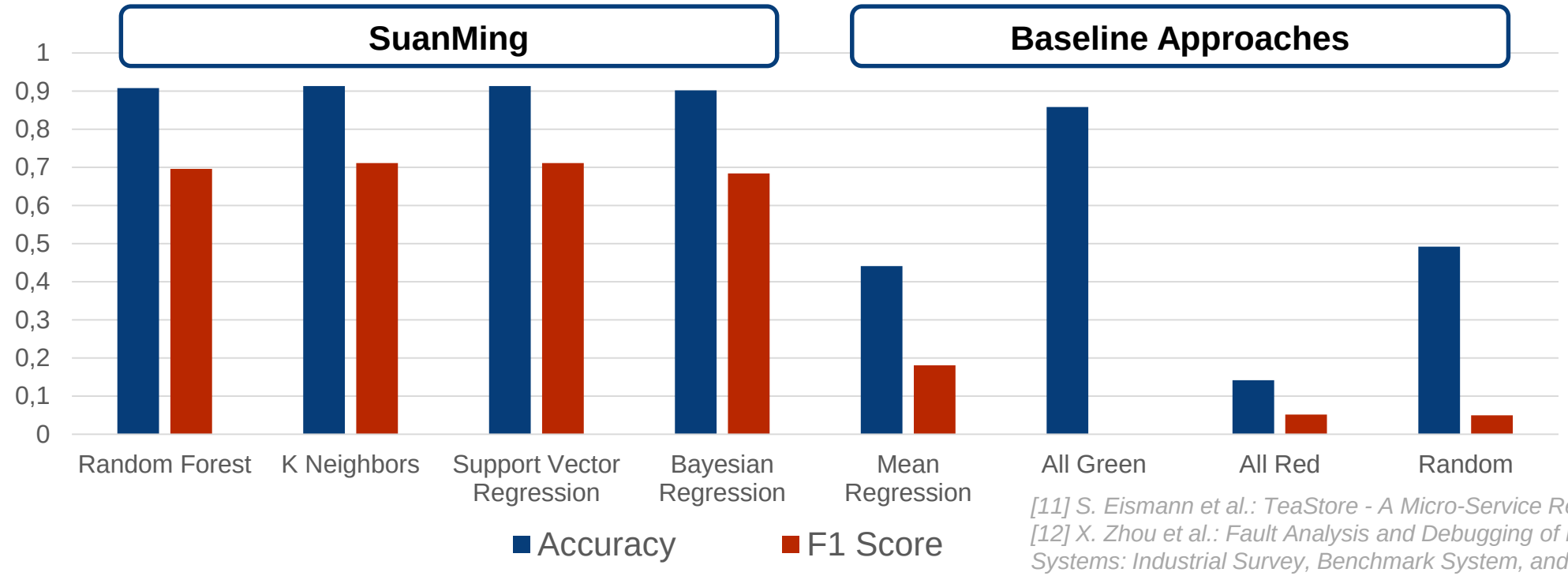
SuanMing: Predicting Performance Degradations

Initialization › *Load Forecasting* › *Request Propagation* › *Performance Inference* › **Root-Cause Detection**



Results

- Tests have been performed both on Teastore [11] and TrainTicket [12] applications
- Deployment in lab and real-world cloud environment
- Example question: Which time is needed to search for a train connection? Does it exceed a fixed threshold?



[11] S. Eismann et al.: TeaStore - A Micro-Service Reference Application
[12] X. Zhou et al.: Fault Analysis and Debugging of Microservice Systems: Industrial Survey, Benchmark System, and Empirical Study

Conclusion

Conclusion

Problem

State-of-the-art reactive APM tools are unable to predict and mitigate performance degradations

Conclusion

Problem

State-of-the-art reactive APM tools are unable to predict and mitigate performance degradations

Idea

Four step prediction process with load forecasting, request propagation, performance inference and root-cause detection

Conclusion

Problem

State-of-the-art reactive APM tools are unable to predict and mitigate performance degradations

Idea

Four step prediction process with load forecasting, request propagation, performance inference and root-cause detection

Benefits

Augments existing monitoring stack, high explainability

Conclusion

Problem

State-of-the-art reactive APM tools are unable to predict and mitigate performance degradations

Idea

Four step prediction process with load forecasting, request propagation, performance inference and root-cause detection

Benefits

Augments existing monitoring stack, high explainability

Future Work

Evaluate more model types, further validation of results

Conclusion

Problem	State-of-the-art reactive APM tools are unable to predict and mitigate performance degradations
Idea	Four step prediction process with load forecasting, request propagation, performance inference and root-cause detection
Benefits	Augments existing monitoring stack, high explainability
Future Work	Evaluate more model types, further validation of results

THANK YOU FOR YOUR ATTENTION