

Usage Tracking Components for Service-Oriented Middleware Systems

Daniel Skrobo, Klemo Vladimir, and Sinisa Srbljic

School of Electrical Engineering and Computing,
University of Zagreb, Croatia

{daniel.skrobo, klemo.vladimir, sinisa.srbljic}@fer.hr

Overview

- Motivation
- Usage Tracking Methods and Systems
- Programmable Internet Environment
- Usage Tracking Services Architecture
- Usage Tracking Languages
- Performance Evaluation
- Conclusion

Motivation

- **Usage Tracking Components**
 - Monitoring of user activities in service-oriented environments
- **High-Level Services**
 - User profiling, security auditing, billing, system health monitoring
- **UT Services Design**
 - Distributed data harvesting
 - Storage space management
 - Usage records aggregation
 - Simplified configuration management
 - Adaptability to various distributed architectures

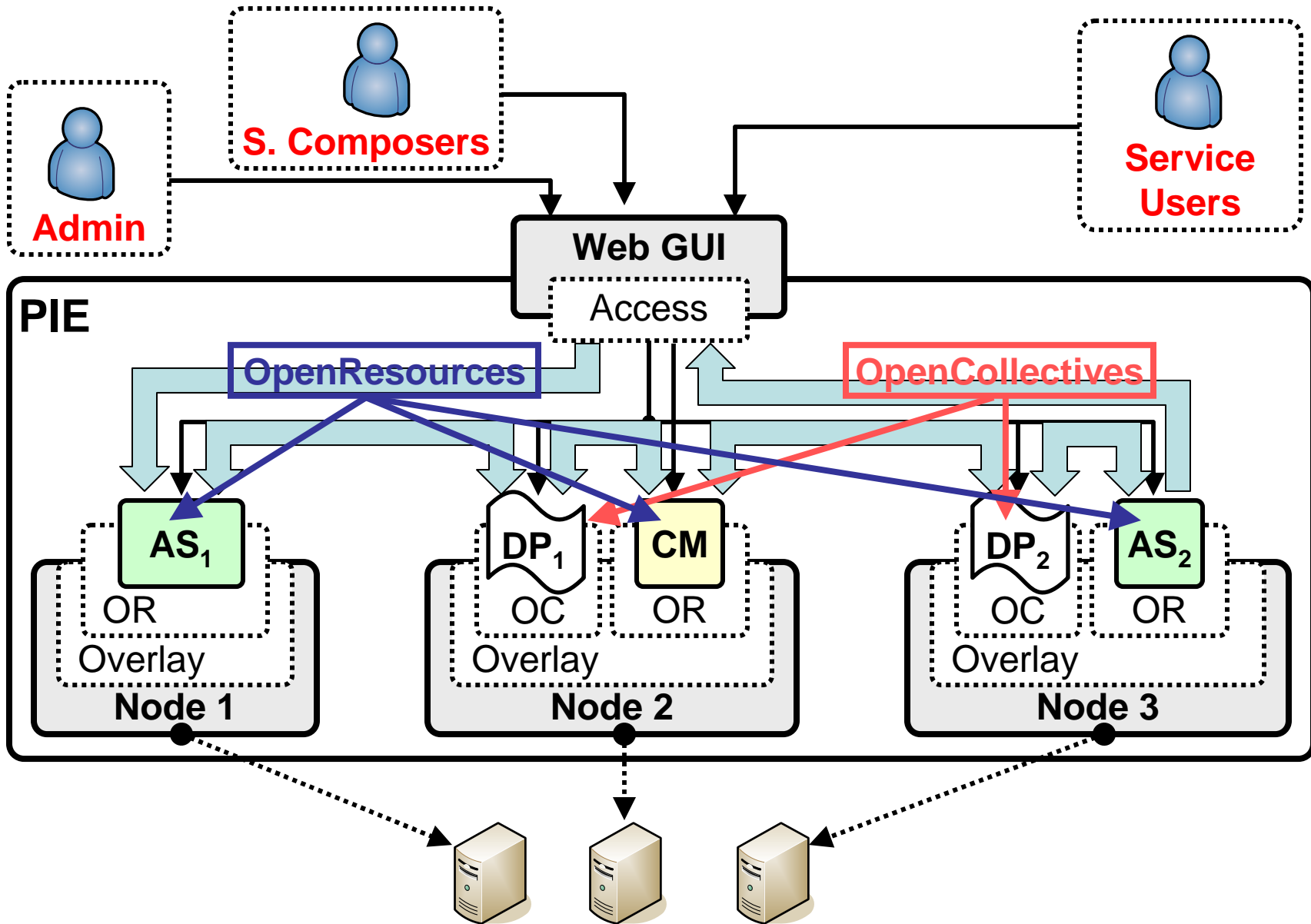
Usage Tracking Methods and Systems

- **Web Server Logs**
- **Grid Monitoring Systems**
 - Ganglia
 - MonALISA
- **Middleware Systems**
 - MidArc
 - Dynasoar
- **View-Based Tracking of Services**

Programmable Internet Environment (PIE)

- **End-User Service Composition Middleware System**
 - Web browser centric design
 - Three-tiered distributed architecture
 - Built on top of WS-* standards
- **Application Development**
 - Composition of Web Services
 - Simple Service Composition Language (SSCL)
- **Cooperation and Funding**
 - Ericsson Nikola Tesla, Croatia
 - Croatian Ministry of Science
- **Presentations**
 - Google, Intel
 - UC Berkely, UC Irvine, Santa Clara University

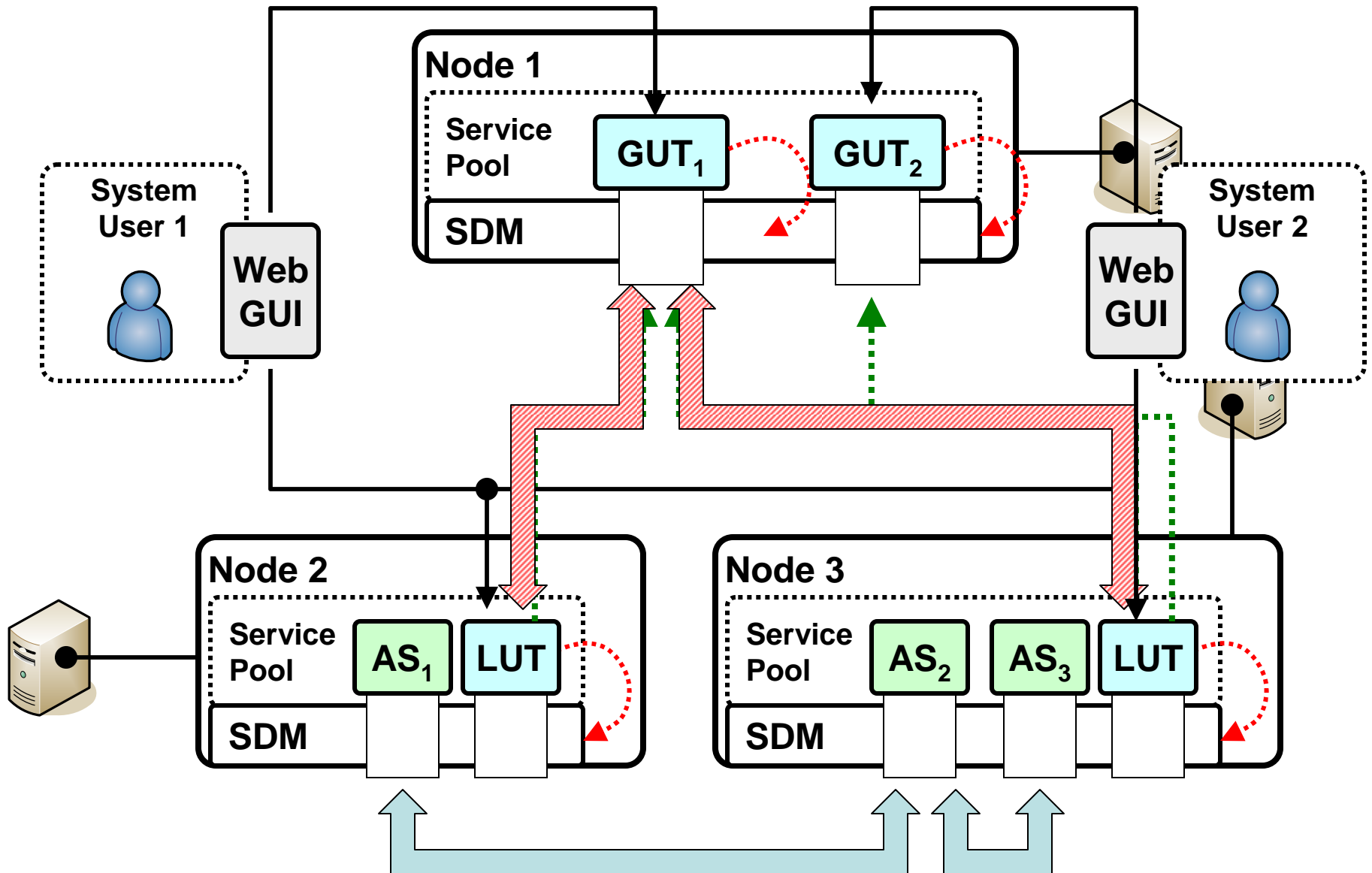
PIE Sample Configuration



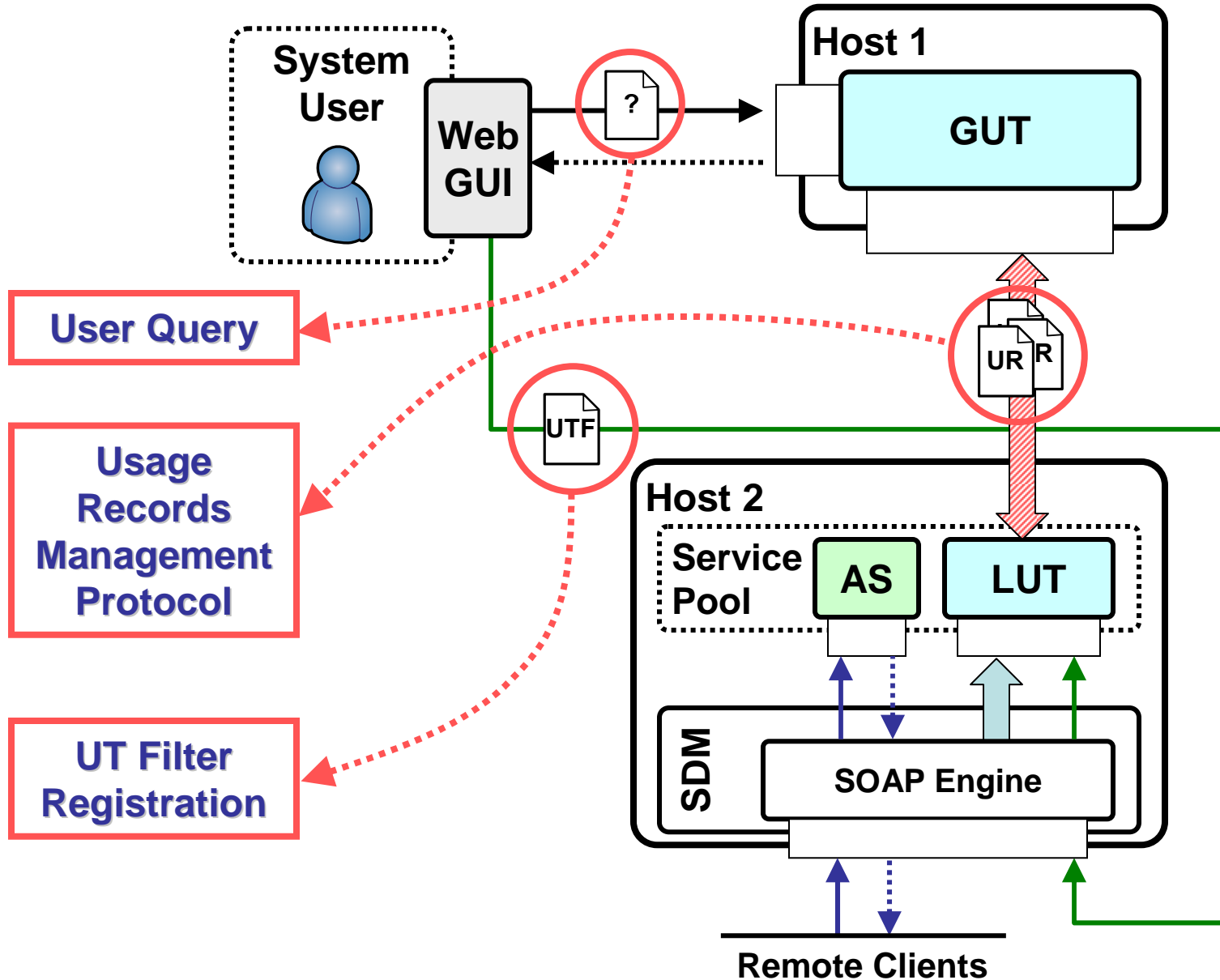
Service-Oriented Usage Tracking Components

- **Usage Tracking Services Architecture**
 - Integration with SOC-enabled systems
 - Adaptability and simplified configuration management
 - Global and local usage tracking services
- **Global Usage Tracking Services**
 - Access to stored usage tracking records
 - Aggregate usage records metadata
 - User query resolving
- **Local Usage Tracking Services**
 - Intercept SOAP traffic
 - Record application service invocation events

UT Services Deployment



UT Services Operation



Usage Records Management Protocols

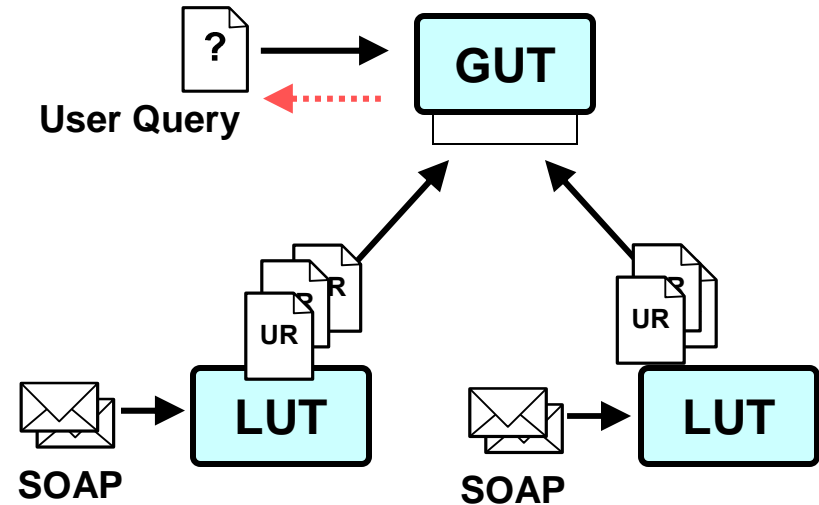
- **Push Protocol**

- **LUT Services**

- Forward UT records to GUT*

- **GUT Services**

- Store UT records*



- **Pull Protocol**

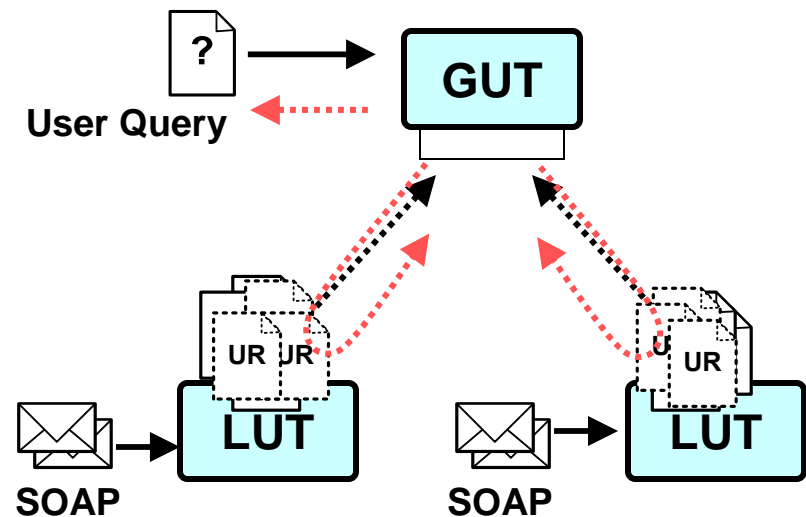
- **LUT Services**

- Locally store UT records*

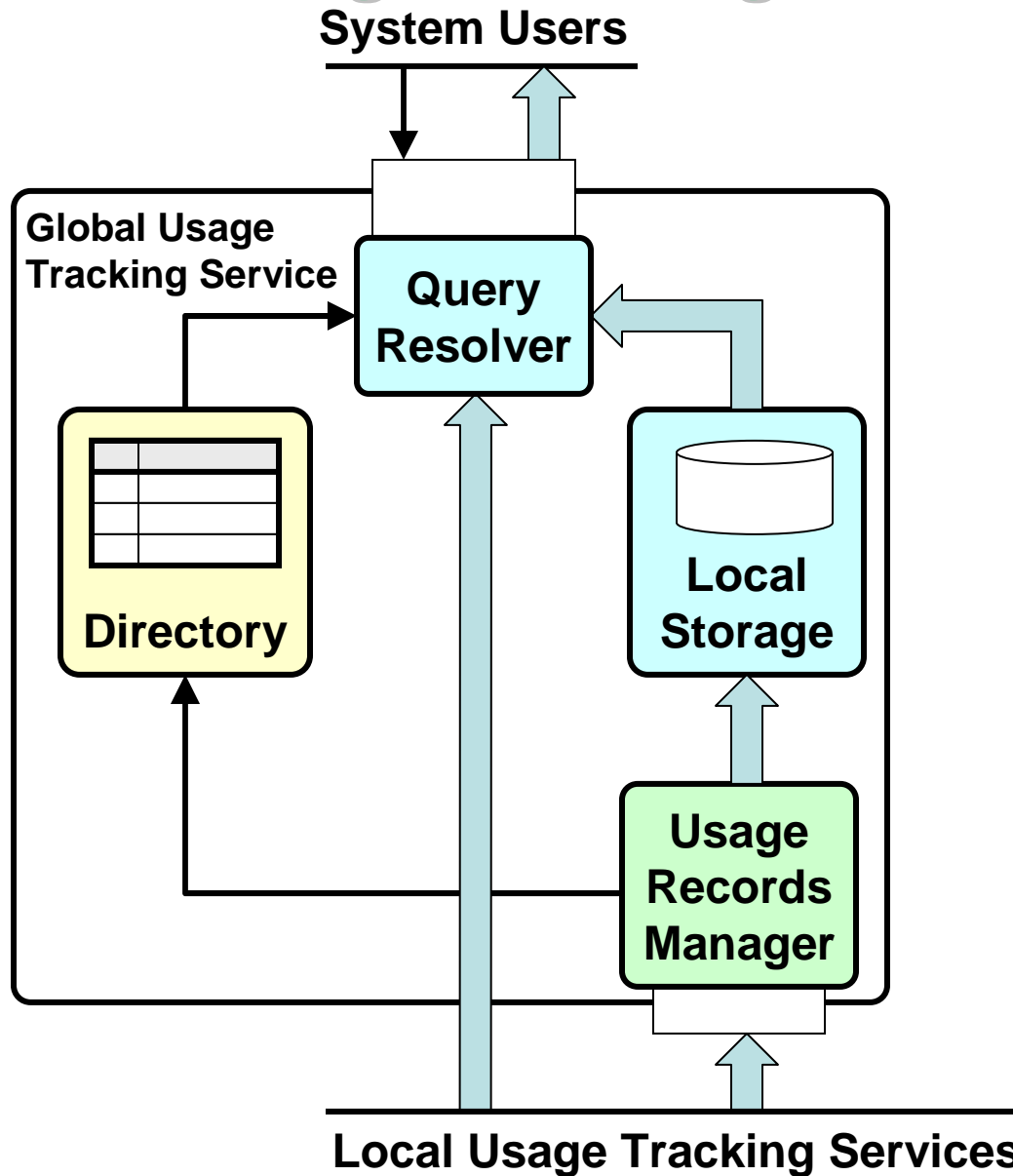
- Send references to GUT*

- **GUT Services**

- Store references UT records*

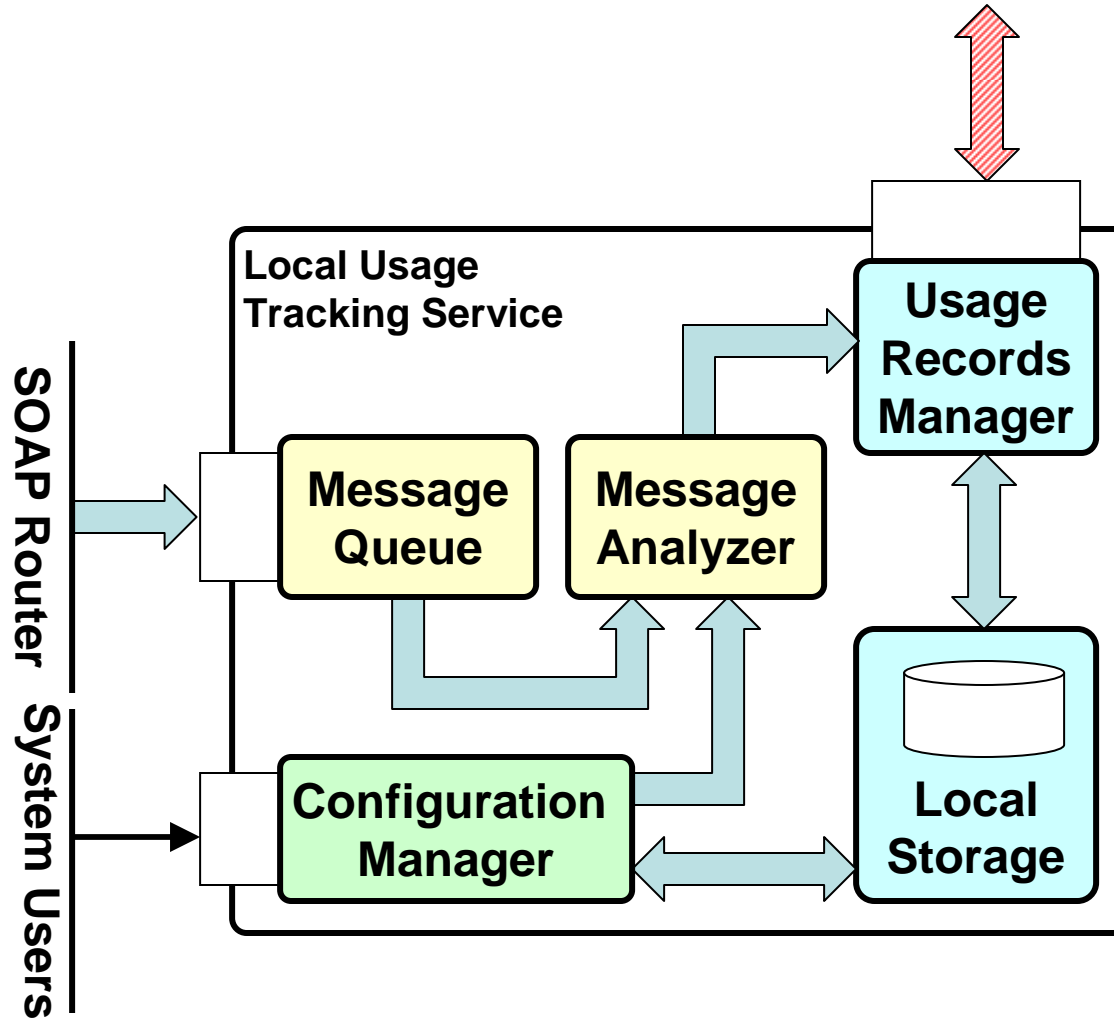


Global Usage Tracking Service

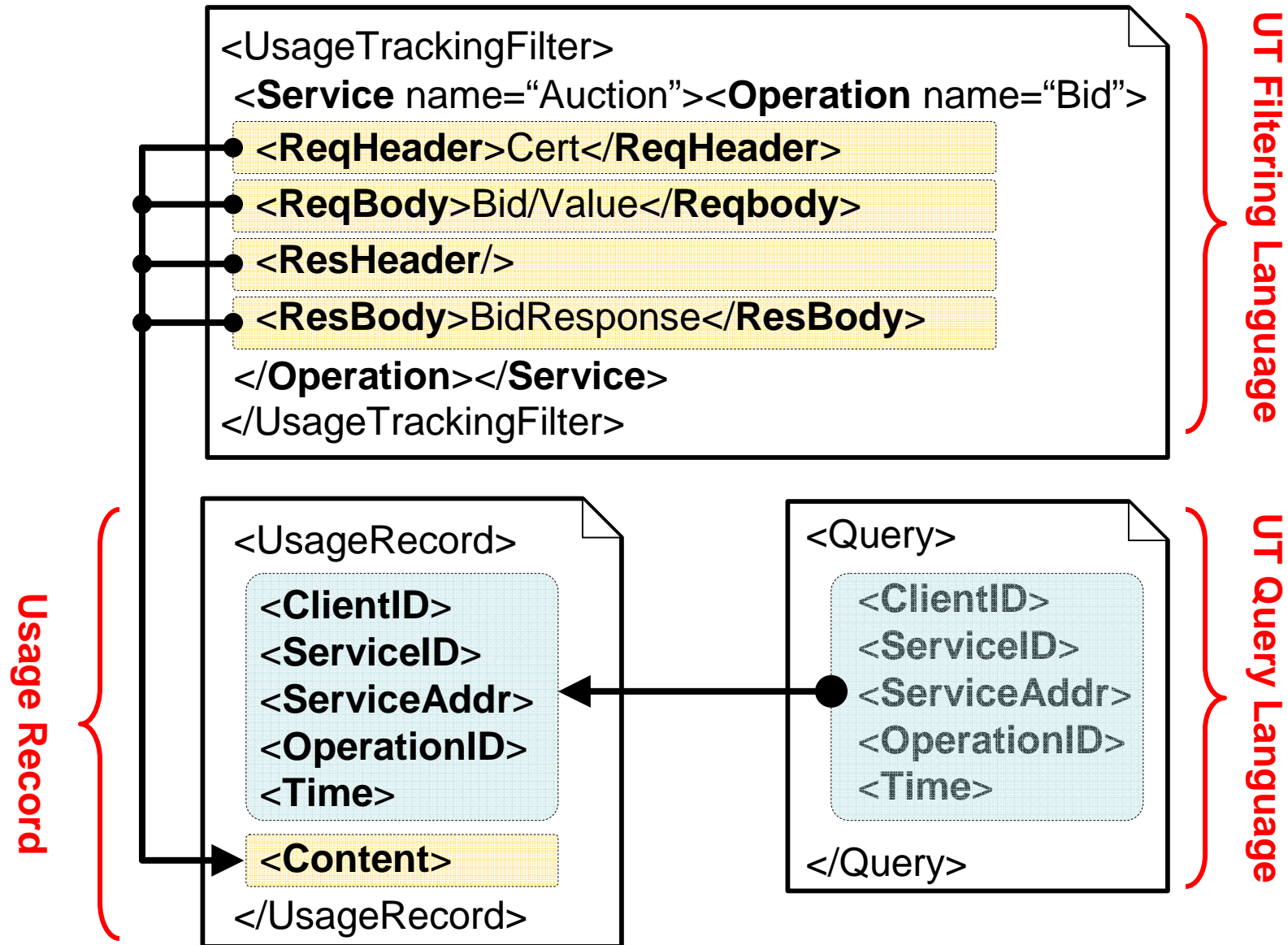


Local Usage Tracking Service

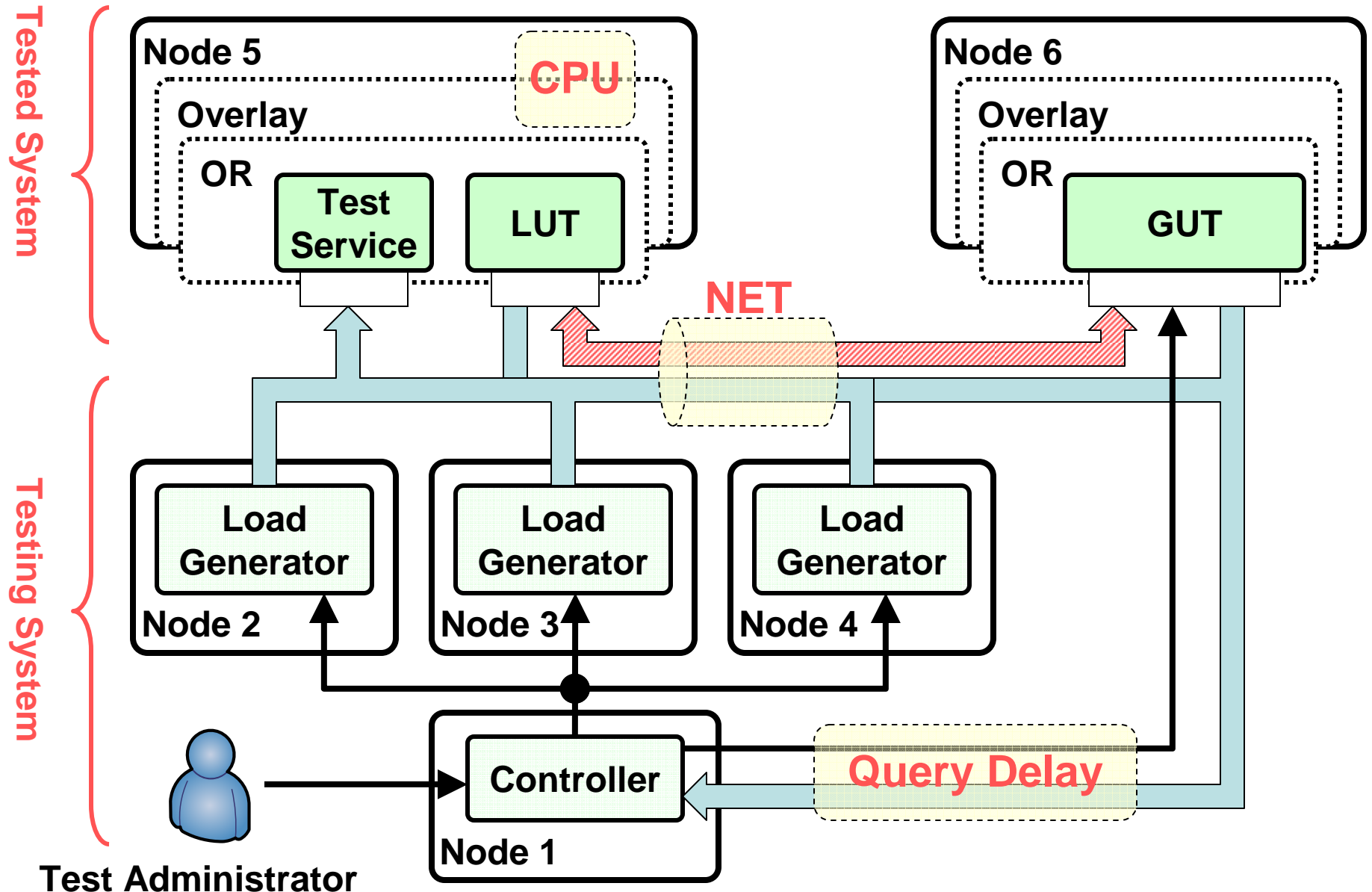
Global Usage Tracking Service



Usage Tracking Languages

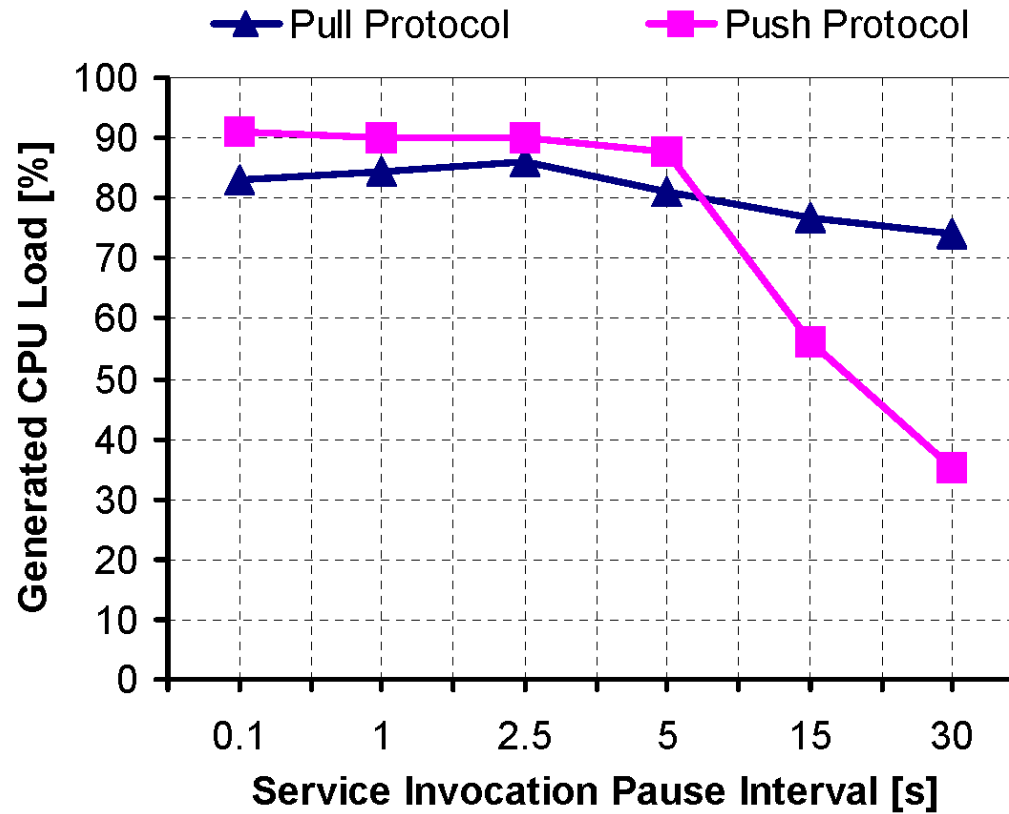


Performance Test Environment



Test Administrator

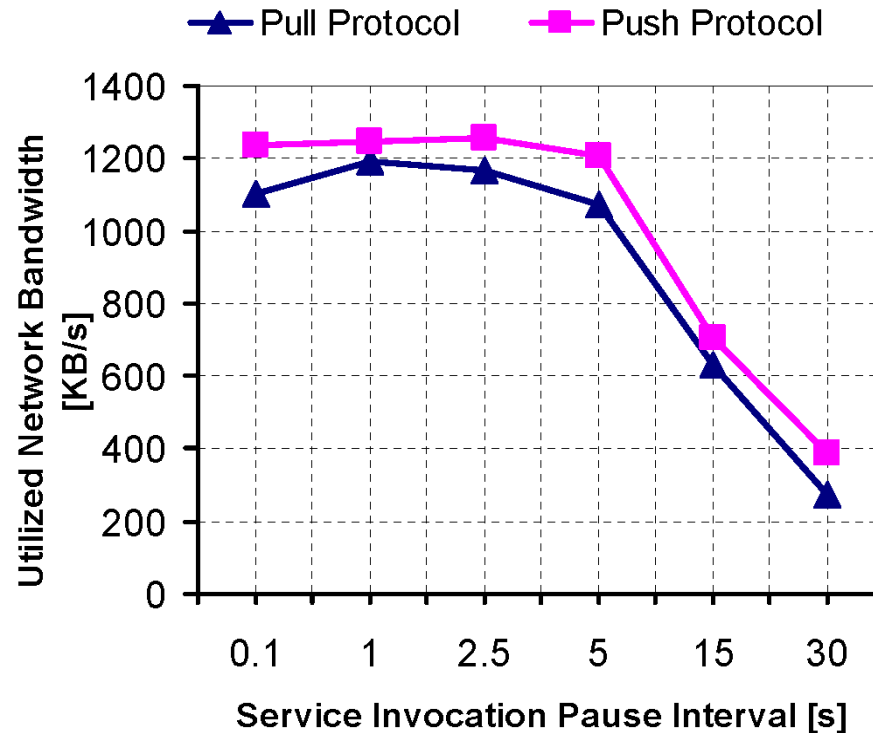
Performance Results



Generated CPU Load in Dependency of Service Invocation Load

Message size: 100 kB

Performance Results



Utilized Network Bandwidth in dependency of Service Invocation Load
Message size: 100 kB

Protocol type	Query resolution delay [s]
Push	0.7 s
Pull	12 s

Query Resolution for Service Invocation Pause Interval of 0.1 s

Conclusion

- **Service-Oriented UT Components**
 - Loosely-coupled collaborative UT services
 - Integration with SOC-enabled systems
 - Adaptability with various distributed architectures
 - XML-based querying and filtering languages
- **Usage Records Management Protocols**
 - Push protocol, Pull protocol
 - Performance evaluation
- **Future Work**
 - Context-aware aggregation of composite service usage records
 - Adaptive hybrid usage record management protocols