

Lessons Learned and Challenges on Benchmarking Publish-Subscribe IoT Platforms

INSTITUIÇÕES ASSOCIADAS



TÉCNICO
LISBOA



universidade
de aveiro



NOKIA



UNIVERSIDADE
BEIRA INTERIOR

U.PORTO

ISCTE IUL
Instituto Universitário de Lisboa



IPL

Instituto Politécnico da Leiria

Ana Aguiar¹, Ricardo Morla²

¹Instituto de Telecomunicações

²INESC TEC



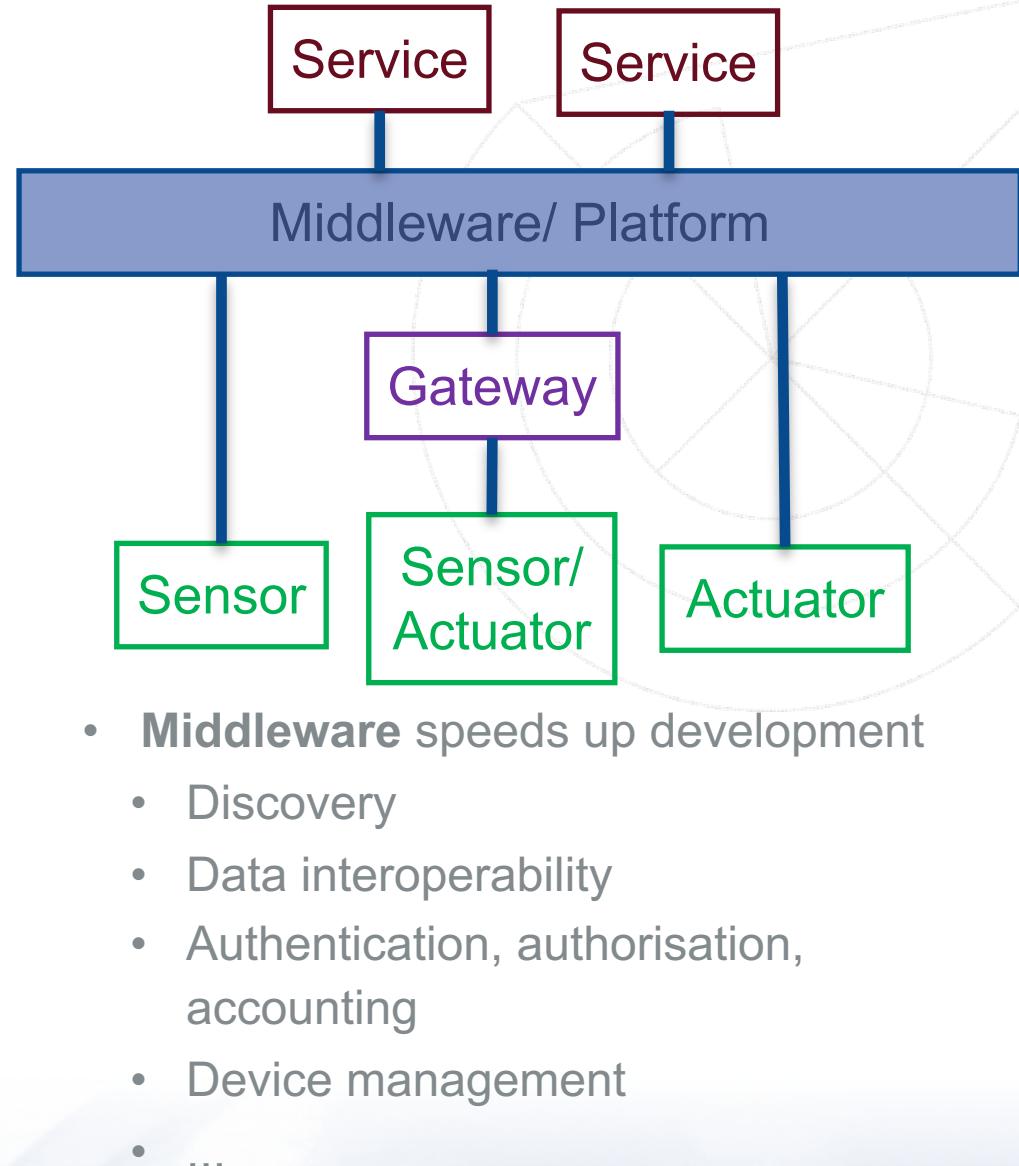
instituto de
telecomunicações

Internet of Things

The **interconnection of devices**, especially low power and small form factor devices, **with services**, to build distributed applications.

Architectural elements

- Sensors and actuators
- Networked data collection
 - Sensor networks
 - Data aggregation, data freshness, caching
- Services, algorithms
 - Data fusion, information extraction
 - Automation



Why benchmark?

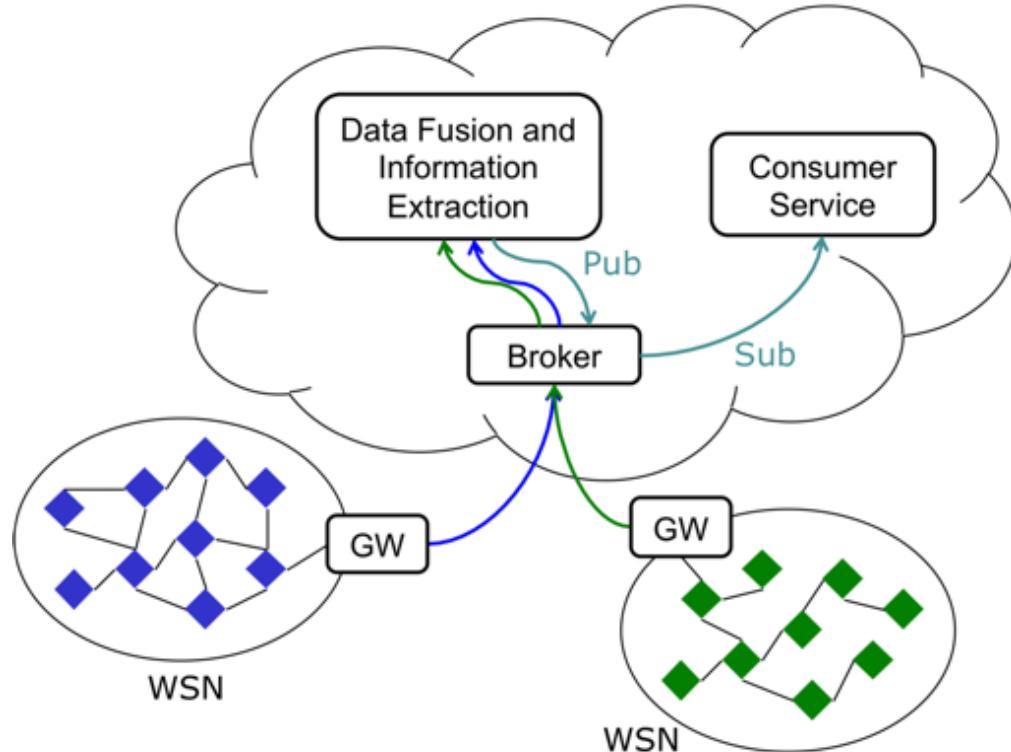
- No comparison grounds for wide variety of platforms/ middlewares
- Decision support beyond feeling and hear-say
- Inform evolution of standards and implementations



vs



Workload: Massive IoT Scenario



Inspired by

- Smart cities
- Intelligent transportation
- Large plants, factories, etc

Use case

- Data points received from multiple sources, e.g. inductive loops, vehicles
- Service fuses data into estimated travel time per graph edge
- Result is published periodically
 - Every 15 minutes
- Consumer service: adaptive routing uses up-to-date travel times

Qualitative vs Quantitative Metrics

Qualitative Metrics

- Functional requirements: IoT-A
 - Protocol and data interoperability
 - Access control, anonymity, security
 - Self-description
 - Query support
- Community support
 - Lively Q&A fora
 - Tutorials
 - Accessible documentation

Quantitative Metrics

- Data overhead
 - Marshalled data
 - TCP payload
 - IP payload
- Publication and notification times
 - Individual
 - Total
- Throughput
- Publications per time unit

“On the wild” vs Controlled Experiments

“On the wild”

- DigitalOcean VM@ London data center running Ubuntu
- FIWARE account on Barcelona’s public broker
- ETSI M2M broker at FEUP and library from Altice Labs
- No control over deployment conditions
- No control over network
 - Network impact difficult to measure
 - RTT
 - HTTP and TCP retries
- Limited to no insights about broker
- Measure actual performance

In the lab

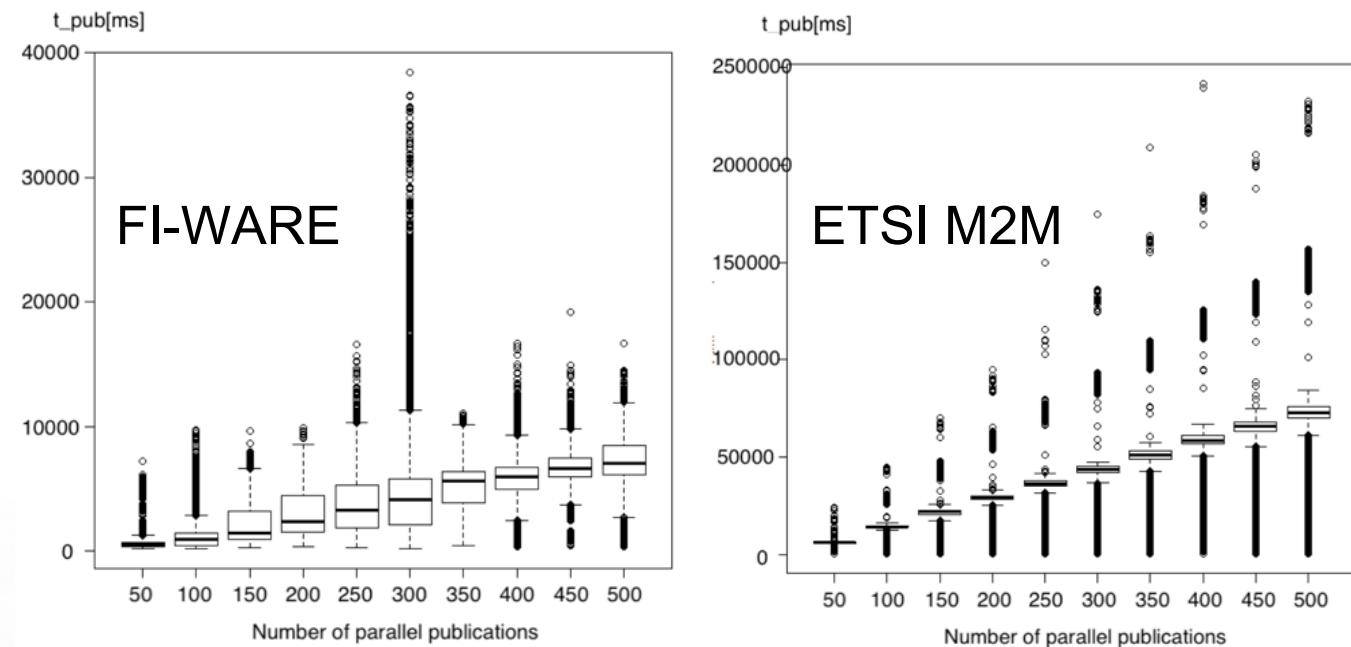
- Broker: Intel Core i5-2500 CPU @ 3.30GHz, 8 GB RAM, running CentOS 6.9
- 100 Mbit/s network connection
- Clients: Intel Core2 Quad CPU Q9300 @ 2.50GHz, 2 GB RAM, running Debian 8.9
- Equivalent system conditions
- Control over network
- Can peek into the broker
 - Measure impact of database
 - Measure impact of resource mapping

Performance Findings: Publication time

- Overall time for publication of a massive dataset is faster if publications are parallelized
- Individual publication times vary largely due to **queueing at the broker**

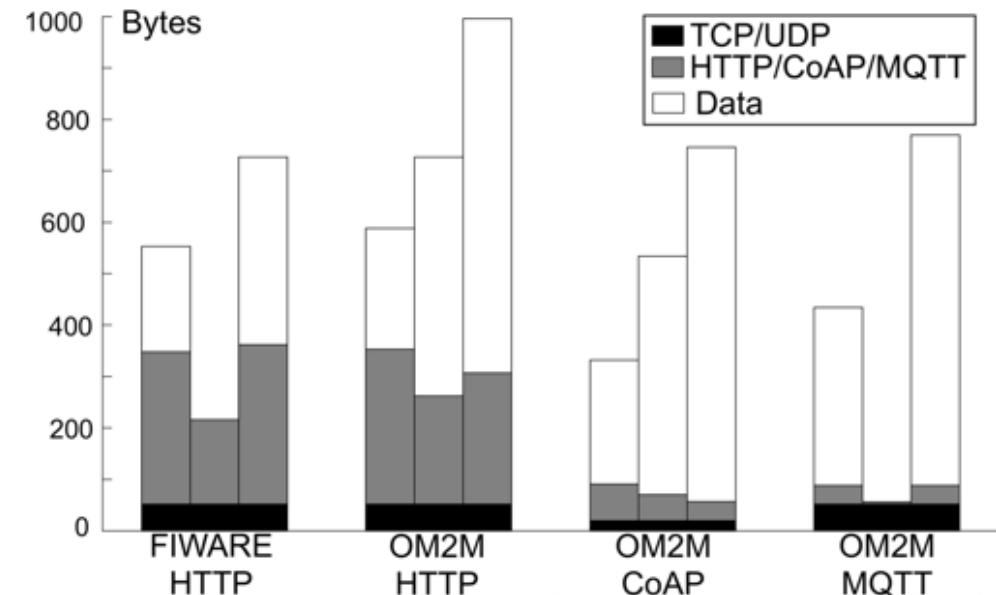
“On the wild” Experiments

	Sequential	Parallel
FI-Ware	64 min	5.42 min
ETSI M2M	98 min	49 min



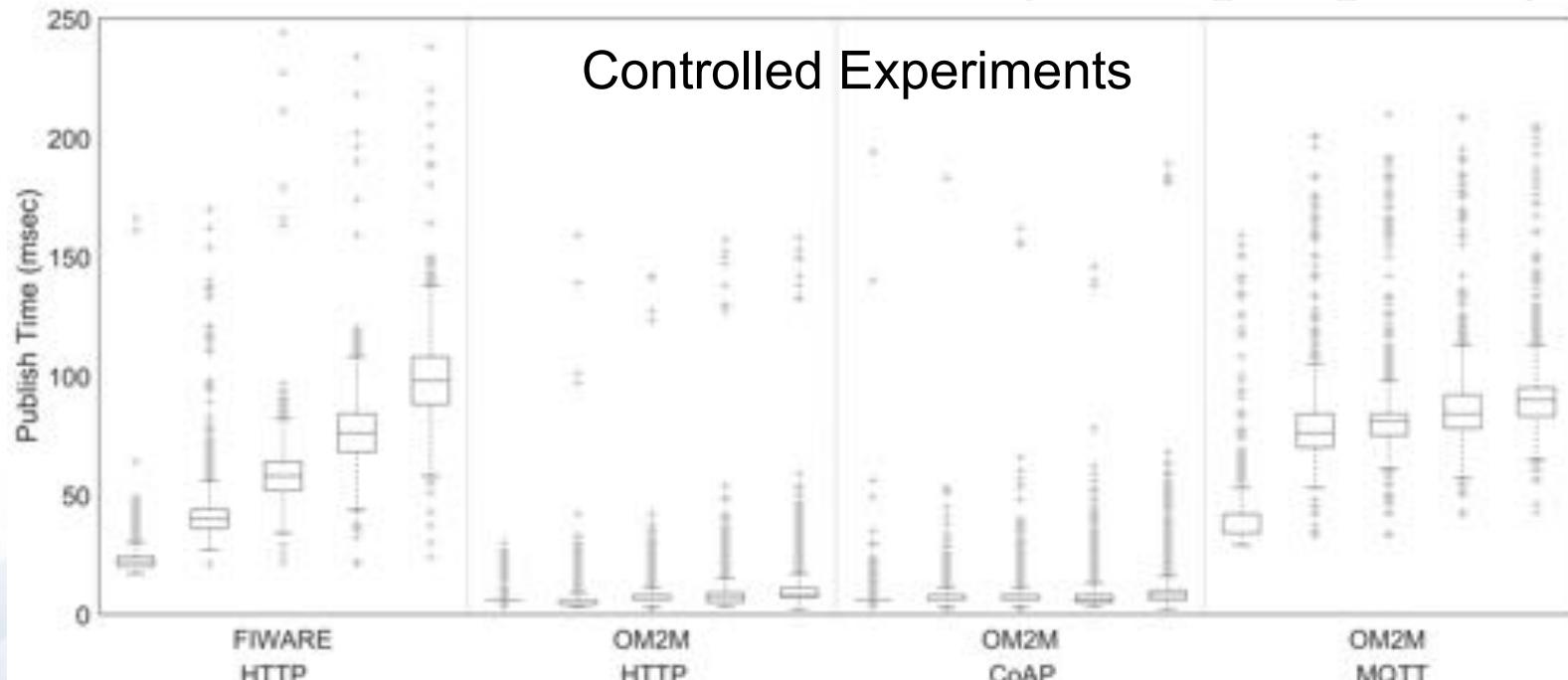
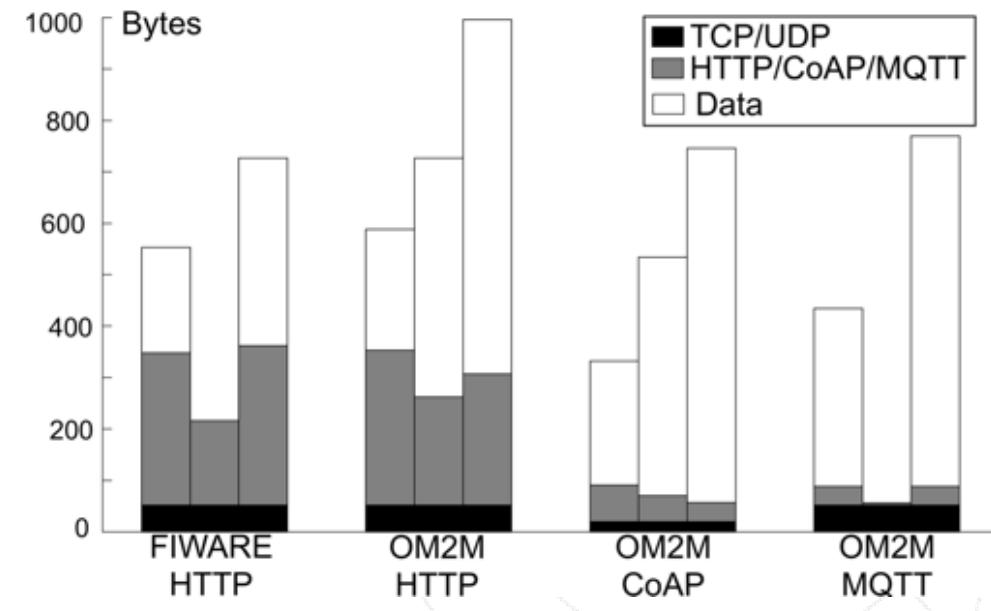
Performance Findings: Overheads

- Communication protocol and data overhead differ significantly
 - HTTP is verbose
 - Implementations may be inefficient
 - Standards may mislead



Performance Findings: Overheads

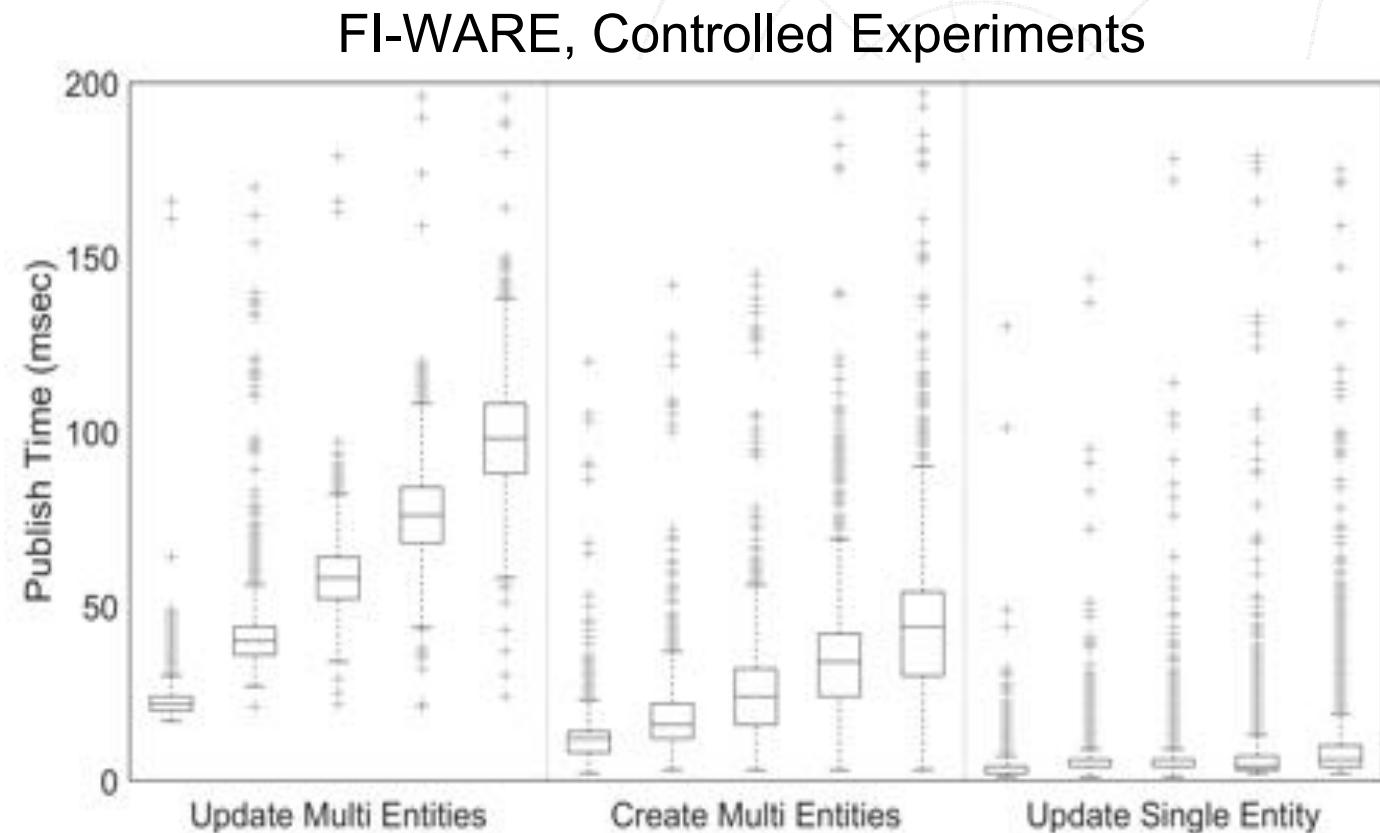
- Communication protocol and data overhead differ significantly
 - HTTP is verbose
 - Implementations may be inefficient
 - Standards may mislead
- Publish time may be at odds with overhead



Controlled Experiments

Performance Findings: Data Organisation

- Overall time for publication of a massive dataset depends on how the **dataset is mapped to data structures**
 - Causes different operations on the broker database
 - One entity per edge**, created beforehand
Publication -> update multiple entities
 - New entity created per publication**
Publication -> create
 - Single entity with edges as attributes**
Publication -> update single entity



Performance Findings: Throughput

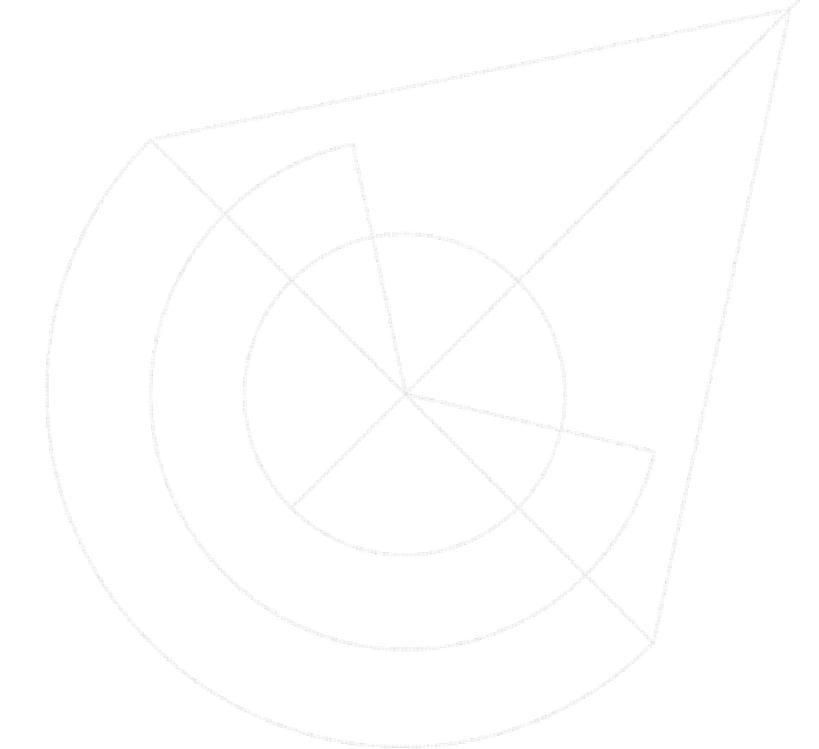
- Measuring the **number of publications per unit time**, e.g. when performing load tests, is challenging
 - "On the wild": how to separate impact of network performance or cross traffic?
 - Controlled: is the result really meaningful?

Performance Findings: Implementation Dependent

- Challenging to distinguish implementation-specific traits from the specification
 - ETSI M2M library only supported 2 TCP connections
 - ETSI M2M library used unnecessary headers
 - ETSI M2M broker implementation implemented synchronous notifications
 - ETSI M2M and OneM2M broker repeated copy of the publication body
 - FI-WARE broker “on the wild” unexpectedly returned 500 responses
 - Cause was authentication module Steelskin PEP
- Infrastructure configuration is very relevant, including network configurations of the used machines and execution environments, e.g. Java version, TCP connections, etc

Performance Findings: Relevant Variables

- Application protocol parameters
 - MQTT QoS or CoAP reliability
 - HTTP headers
 - Marshalling
- Database state and resource mapping
- Execution environment and network configuration
 - Client software, e.g. JVM version
 - Network configuration, maximum number of TCP connections
 - Specific libraries used
- Network conditions if “on the wild”
 - Latency at measurement time
 - Different hours of day, days of week, weeks, where does it end?
 - HTTP retries, TCP RST



Learnt Lessons about Performance

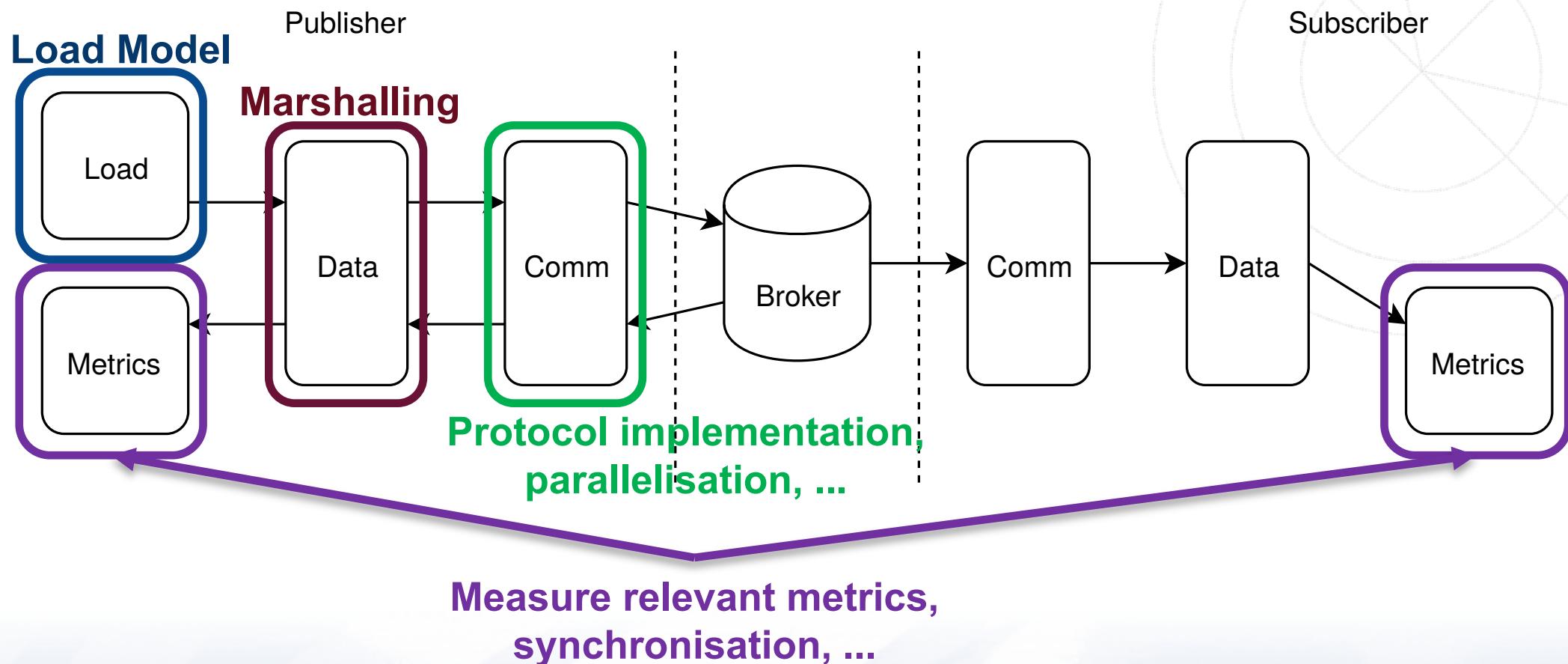
- **Communication protocol and data overhead** differ significantly across platforms
 - IP payload traffic can be 20 times the actual data to be published
- Overall time for publication of a massive dataset
 - is faster if publications are **parallelized**
 - depends on how the **dataset is mapped to data structures**
- **Measuring** the number of publications per unit time is very challenging
- Identified **relevant variables** for benchmarking

- Benchmarking is useful for testing a specific implementation and deployment
- Absolute statements should not be expected, even for same workload and metrics

Learnt Lessons about the Broker

- **Broker** is a key component of high complexity with significant impact on system performance
- Measuring the **broker as a black box** has many caveats
 - Broker is a specific instantiation
 - Only speculation about causes of observed behavior
- **Broker database** is one major bottleneck
 - Each publication is an operation on the database
 - Performance depends on database status, e.g. number of lines

A Modular Tool for Benchmarking Brokers

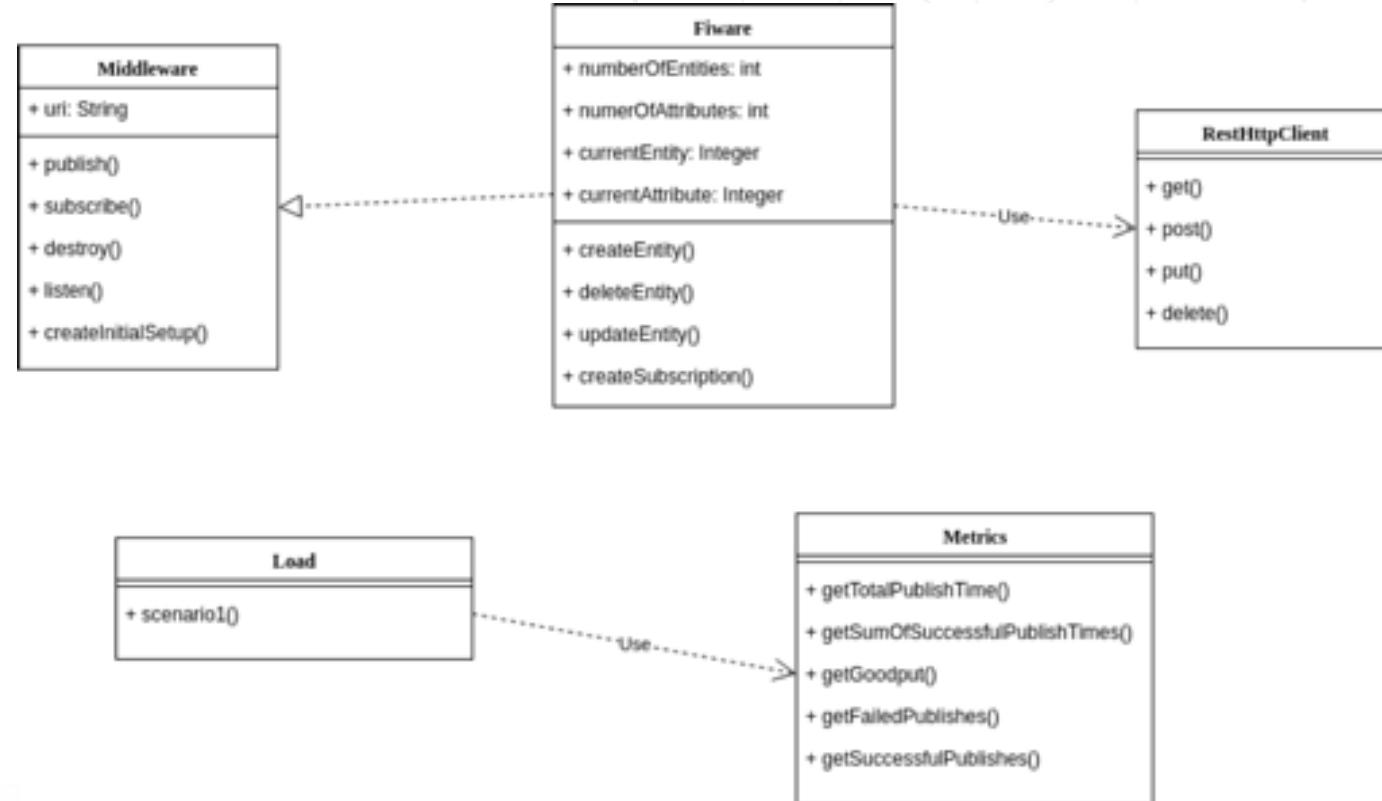


A Modular Tool for Benchmarking Brokers

Class **Middleware** enforces same operations

Classes **Load**, **Metrics** enforce same experiment conditions

Protocol class ensures same protocol versions are used



A Modular Tool for Benchmarking Brokers

Class **Middleware** enforces same operations

Classes **Load**, **Metrics** enforce same
experiment conditions

Protocol class ensures same protocol
versions are used

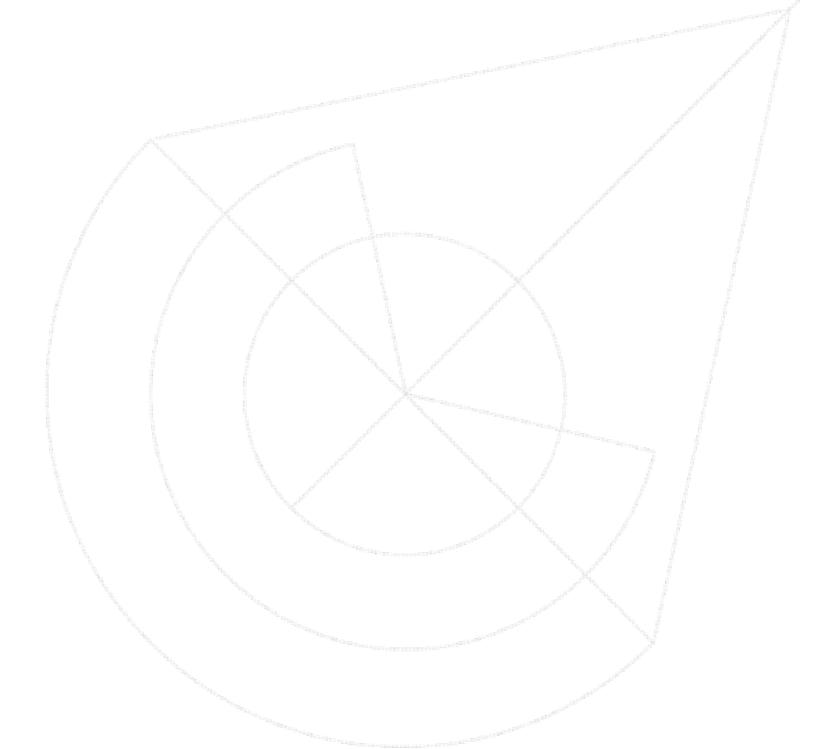
Increases implementation efficiency by
lowering effort for experimenting new
middleware or protocols

Still a lot of work to be done

Class	Lines
Load	119
Middleware	16
Metrics	136
OneM2M	198
FI-Ware	193
Ponte	88

Challenges

- Vary workload qualitatively and quantitatively
 - Adequate models?
- Impact for energy constrained nodes
 - Via energy consumption models
 - Via testbeds
- Contemplate security
 - Impact of known vulnerabilities
 - Security overhead
 - Vulnerability to side-channel attacks
- ...



Conclusions

Controlled experiments are useful for fully comparable circumstances

“On the wild” experiment results may be polluted by networking artifacts and computer system configurations

Identify set of metrics and relevant variables

Broker is a key component of high complexity.

We propose a tool to provide comparability and repeatability of benchmarking experiments

Thank you. Questions?