Enabling Data Exchange in Smart Spaces

“what is the occupancy of the room 2065?
“decrease the temperature of those rooms with occupancy above 50% of their capacity?”
**IoT heterogeneity at multiple layers**

![Diagram of IoT layers]

**Middleware protocols in the mobile IoT**

- DPWS
- CoAP
- MQTT
- ZeroMQ
- WebSockets

- Client-server
- Pub/sub
- Streaming

Towards End-to-end Data Exchange in the IoT – Georgios Bouloukakis
Heterogeneous interconnections in the mobile IoT

How to enable interconnections in the mobile IoT?

Our proposed solution

Automated synthesis of interoperability artifacts:
• enables functional middleware-layer interoperability

Automated placement and deployment at the Edge:
• enables the deployment of interoperability artifacts at the Edge
Models for core interaction paradigms

Client–Service (CS)
- Tight Time & Space Coupling

Publish-Subscribe (PS)
- Time & Space Decoupling

Data Streaming (DS)
- Tight Time & Space Coupling

Tuple Space (TS)
- Time & Space Decoupling

Data eXchange (DeX) connector model

Our generic connector defines 4 basic interaction types:

- one-way
- two-way async
- two-way sync
- two-way stream

Each interaction is represented as a combination of post and get primitives.

We rely on the DeX abstraction to introduce our middleware protocol interoperability solution.
Our middleware protocol interoperability solution

Data eXchange Mediator Synthesizer (DeXMS)\(^1,2\)

- Mediator architecture: relies on DeX for automated Mediator synthesis
- Primitives & data conversion between the common protocol and the Things’ protocols
- A universal way to describe the Things’ I/O required

\(^1\) G. Bouloukakis et al., FGCS, 2019
\(^2\) G. Bouloukakis et al., ICSOC, 2016

Automated Mediator synthesis

Generic Interface Description Language (DeXIDL) & Generic Mediator
The *Where* and *How* Problem

- *Where* to place mediators?
  - Cloud, Edge, and Fog Computing

- *How* to place mediators?
  - Related Problem: Operator Placement
  - Compute a “cost space”\(^1\) to represent *Things* and *Physical Nodes*
    - E.g., a smart building with heterogeneous Things
  - Place mediators in an optimized manner

Criteria: distance, energy, bandwidth, etc

Optimization techniques\(^2\): constraint programming solvers, heuristics, linear programming, bandwidth, etc

---

**Where**: The Cloud, Edge and Fog

- Obvious solution: The Edge and Fog
  - Things push data to the cloud to be analyzed (e.g., 4k camera)
    - Use artifacts at the Edge/Fog to filter these data
  - Timeliness, data privacy, etc

- Work in progress:
  - Systematic solution to automate the deployment of mediators at the Edge
  - Utilize this solution to deploy mediators and other artifacts in the I3 platform

\(^1\) P. Pietzuch et al., ICDE, 2006
\(^2\) V. Issarny et al., ICDCS, 2019
Mediators at the Edge

Docker
- Delivers software in packages called containers.
- DeXMS provides mediators as Dockerfiles.
- A Dockerfile produces a Docker image.

Ansible
- Automation tool to perform installation, maintenance, or monitoring operations.
- Used to automate the installation of the smart space infrastructure as well the DeXMS service.

DeXMS
- Triggers the DeXMS service to generate the required Mediator containers.
- Can be used to monitor networks and services.

Kubernetes
- Container-orchestration system for automated deployment, scaling, and management.
- Supports the automated deployment of mediators.

DeXMS novelty

- Lightweight architecture
- Mediators employed only when necessary
- Any common protocol
- Support for any protocol classified under CS, PS, DS & TS
- Evolution support
- Automated Mediator synthesis
- 75-96% person-hours reduction when using DeXMS
- Work in progress: enabling application-layer data exchange

---

1 R. Yus et al., Buildsys, 2019
Software artifacts and adoption

- DeXMS is part of the zefxis¹ platform (https://gitlab.inria.fr/zefxis):
  - Mediator generator: https://gitlab.inria.fr/zefxis/DeXMS
  - Eclipse plugin for defining Things' DeXIDs: https://gitlab.inria.fr/zefxis/DeX-IDL
  - Web console: https://gitlab.inria.fr/zefxis/IoT-web-console

- Demos:
  - Mediator generation: https://youtu.be/UgfM3810RS8
  - Web console installation: https://youtu.be/I6jZ5u3QYOw
  - Fire Detection scenario: https://youtu.be/SJeiqJkBhls

- DeXMS is used as a core component in H2020 CHOREVOLUTION and Inria/UCI MINES projects.

Thank you!