Low level architecture of the PERLA middleware

PerLa
PERvasive LAnguage

Andrea Maesani 719697
Claudio Magni 720827
Emanuele Padula 719348

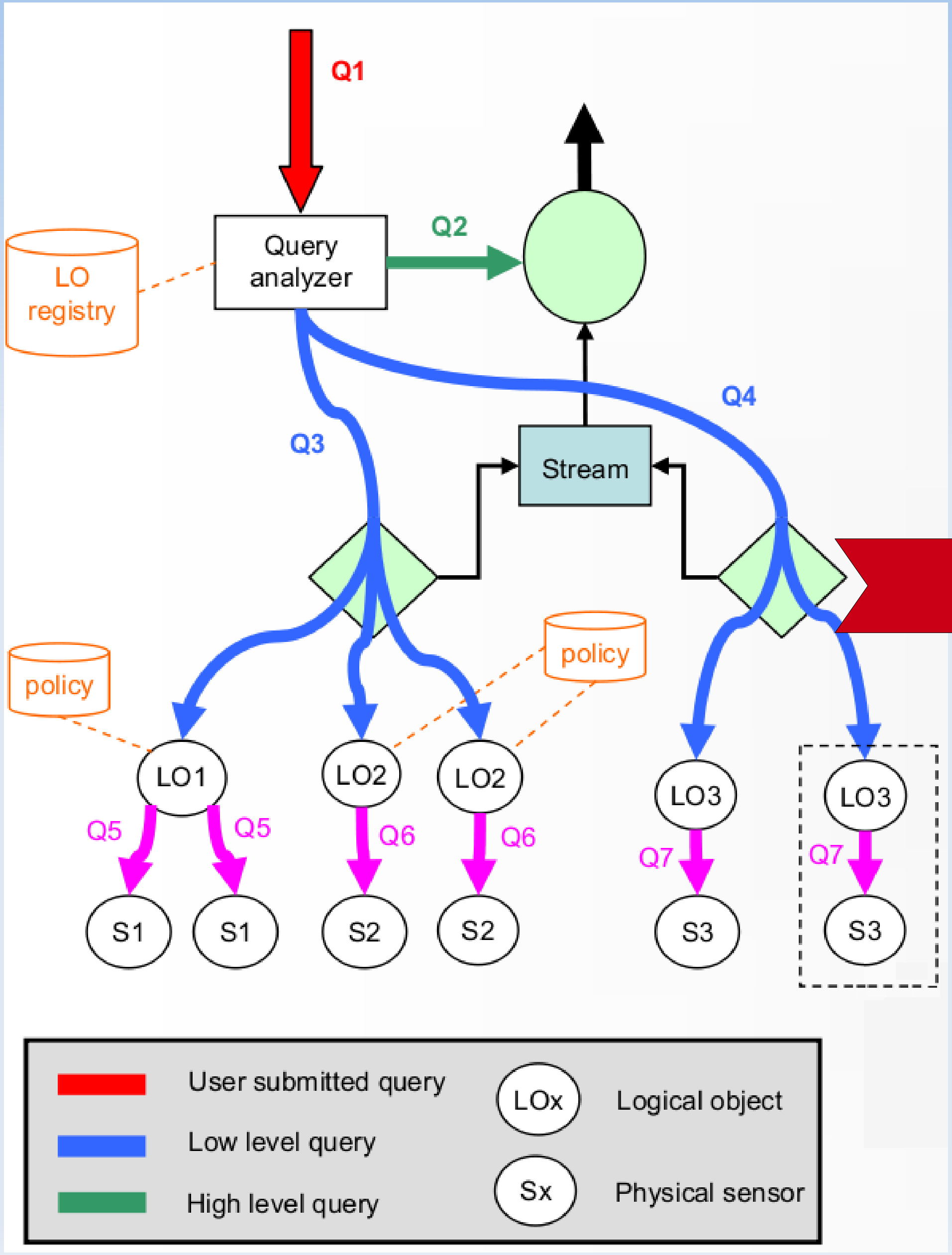
Outline

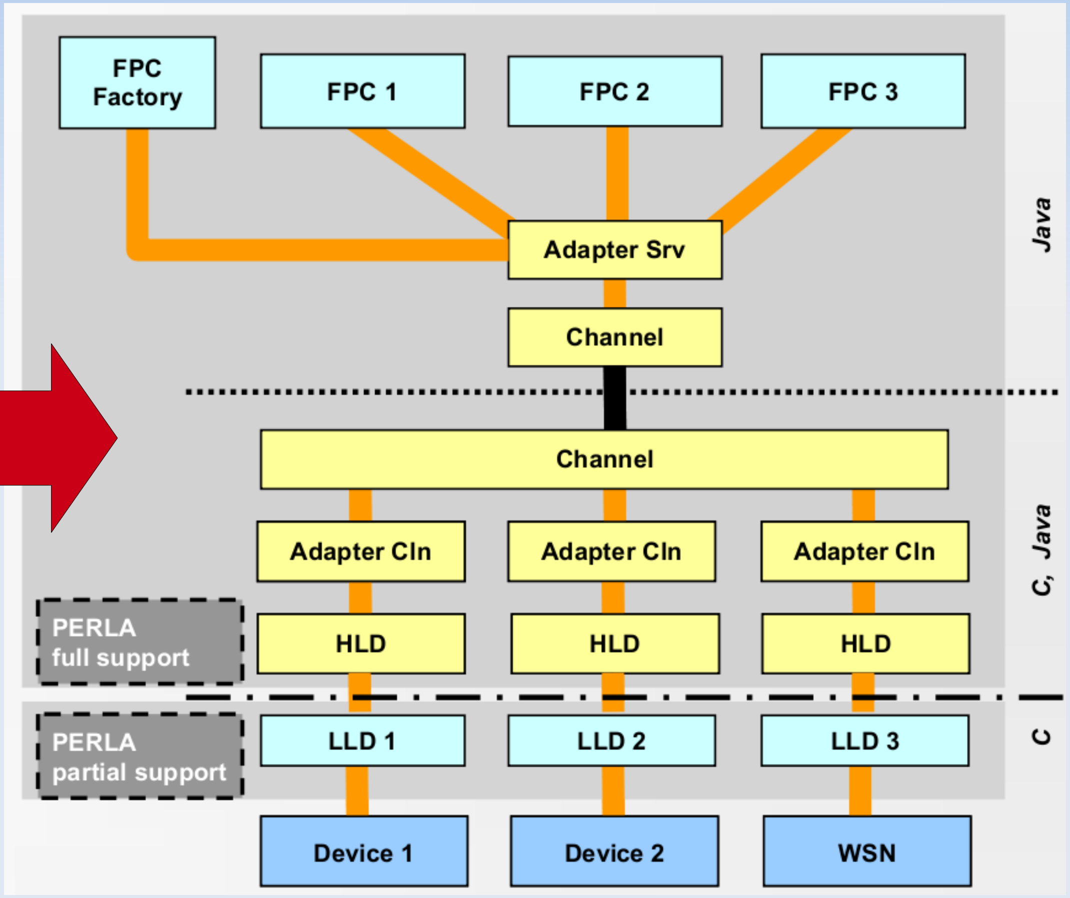
- Introduction to PERLA
  - Middleware overview
  - Low level architecture
- Project's aim
- The low level architecture stack
  - FPC Factory
  - Adapter level
  - Channel level
- Additional features developed
- Network testing

Introduction to PERLA

- Full declarative **SQL-like** high level language to query **pervasive systems** hiding the complexity of handling **different technologies**

Middleware's general overview

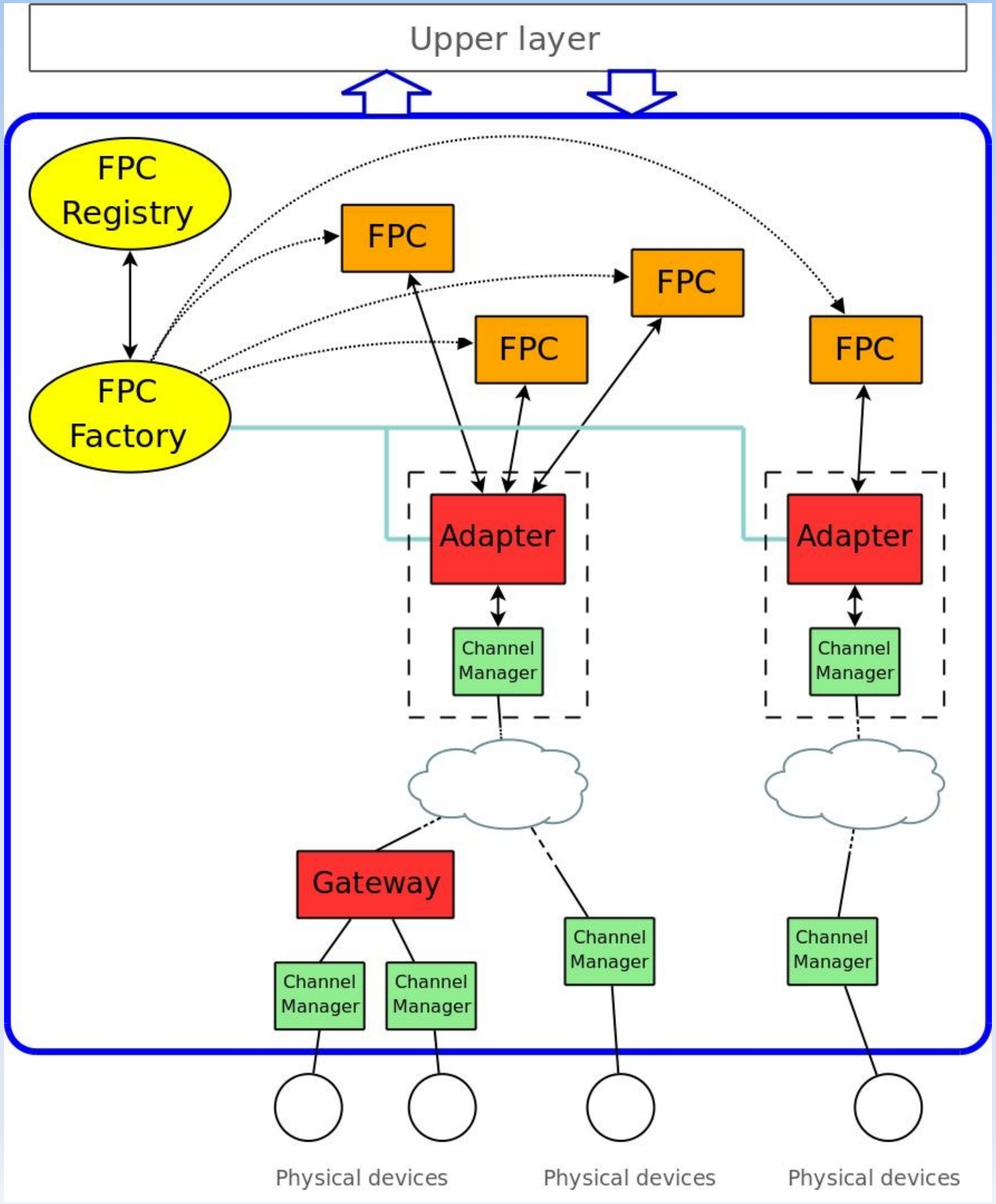




Project's aim

- Implementation of the low level PERLA architecture supporting:
  - Channel Level
  - Adapter Level
  - Message routing/relaying
  - Physical device binding

Low level architecture



FPC Factory

- Functionality Proxy Component
  - It's the logical object that abstracts a device
- Requirements
  - Creates logical objects
  - Strictly bound to the Registry
  - Not much more than a stub at the moment

Adapter level

- Requirements
  - Routes messages from a logical object to the relative device and viceversa
  - Handles Virtual Channels
- Two kinds of middleware machine
  - Standard middleware machine
    - Handles normal communication
    - Involved in the binding procedure at device start-up
  - Gateways Machines
    - Relays messages between heterogenous networks

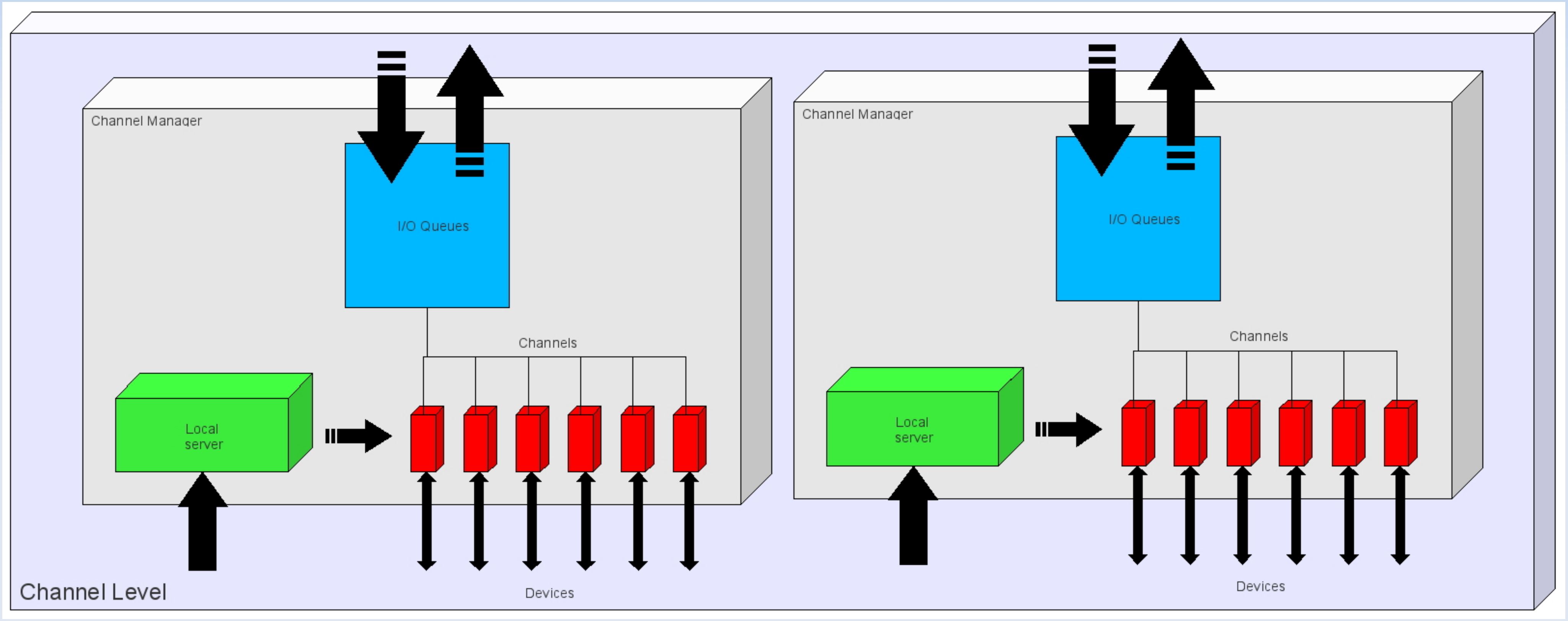
Binding procedure

- A new device sends a binding message:
  Binding flag + BVCI + XML descriptor
- The adapter checks whether the device can continue the binding procedure
- The adapter forwards the message to the relative FPC factory
- The FPC factory creates a new FPC for the device
- The new FPC sends an ack message
- The adapter server creates a new vci and sends to the device the ack message
  Ack flag + BVCI (old) + VCI (new)

Channel Level

- Channel manager
  - An abstraction to the underlying physical channel [socket, console, serial]
  - Provides a simple way to use the channel to upper levels
    - read(addr, payload)
    - write(addr, payload)
  - Supports addressless/addressful channels
  - Keeps track of physical devices to support network failures/device disconnections
- Channel
  - A bidirectional channel between two machines

Channels Implementation Details

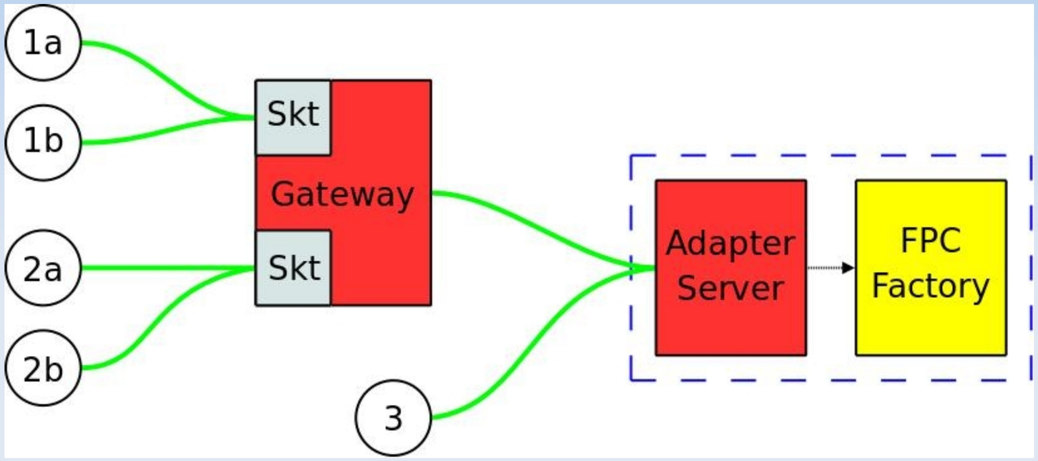


Additional features developed

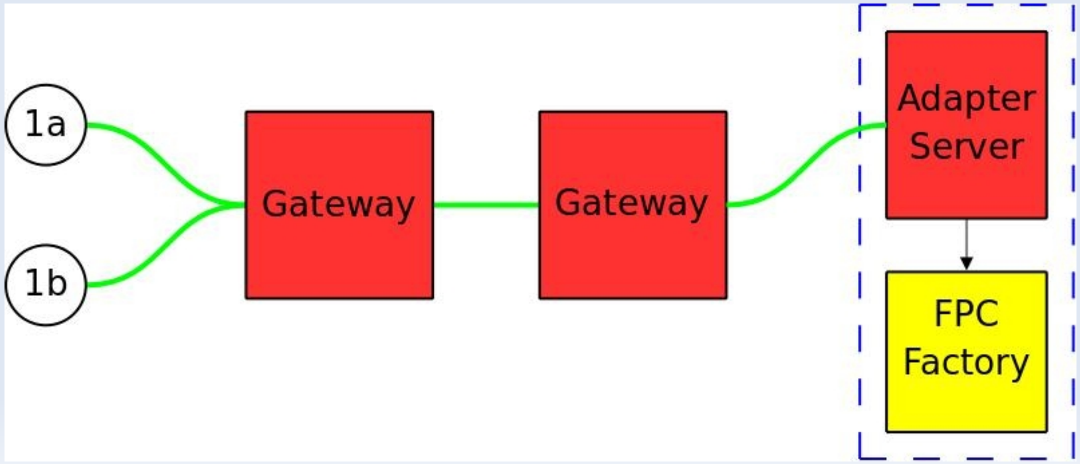
- Header support to the channel level frames for future expansions
- Initial support for channel failures management
- An XML based automatic configurator for each middleware node

Network testing

Generic network test



Gateway test


1a
1b
Gateway
Gateway
Adapter Server
FPC Factory