Application Services Backbone Choices

- Message Bus
  - Generally standardized; avoiding vendor lock in - many providers
  - Services publish/consume messages
    - Always on technology; single scale
    - Scale at the endpoints
  - Plenty of language bindings
  - Time trusted technology
  - Testing/debugging at the endpoints

- FAAS/Lambda/Serveless Technology
  - 3rd party / vendor provided – lock in concerns
  - Language runtimes may be limited
    - Many are built in Go
    - Typically packaged as Docker container
  - Micro-micro services are brought into FAAS provider when needed
    - Only on when needed; scale per function need
  - Early adopter technology; largely cloud platform based
  - No standardized security built in
  - Can be hard to test/debug
  - Not ready for edge (size, deployment options, etc.)

Message Bus Options

- Message Broker
  - Mosquitto
  - Paho
  - Apache ActiveMQ
  - Apache Apollo
  - RabbitMQ
  - Qpid
  - NATS
- No Message Broker
  - 0MQ
  - Nanomsg
  - GRPC
  - Thrift

FAAS Options

- Need non-Cloud provider
- OpenFaas
- OpenWhish
- Kubeless
- IronFunctions
- nuclio

Recommendation

- FAAS/Serveless probably too early
- Not stable yet
- Not edge ready
- Can we build functions in a way that supports FAAS in the future
- Architect the service functions small and well defined so that some day they could be moved to a FAAS framework with little work

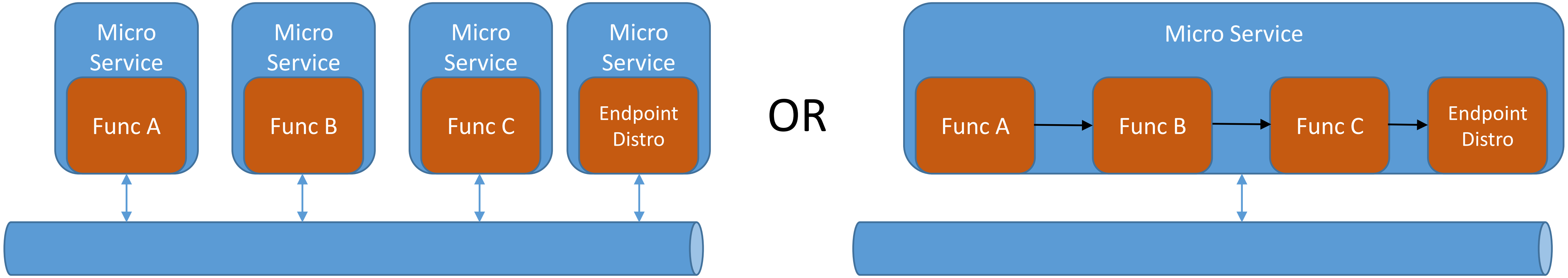
Application Services Functions

- Essentially an EAI engine
- Operations
  - Filter (only give me readings from device A; only give me readings regarding temperature, …)
  - Validator (device ID, reading against value descriptor, …)
  - Transformation (convert C to F values, convert CBOR to Protobuf, …)
  - Enrich (add device metadata to reading, …)
  - Format (JSON, XML, CSV, …)
  - Encrypt (really different kind of transformation)
  - Compress (really different kind of transformation)
  - Custom (black box that you define what you want to happen inside)

Application Services Functions

- Endpoints
  - HTTP(s)
  - MQTT(s)
  - AMQP
  - XMPP
  - WebSockets
  - CoAP
  - Cloud providers
    - Azure IoT
    - AWS IoT
    - Google Cloud IoT
    - …

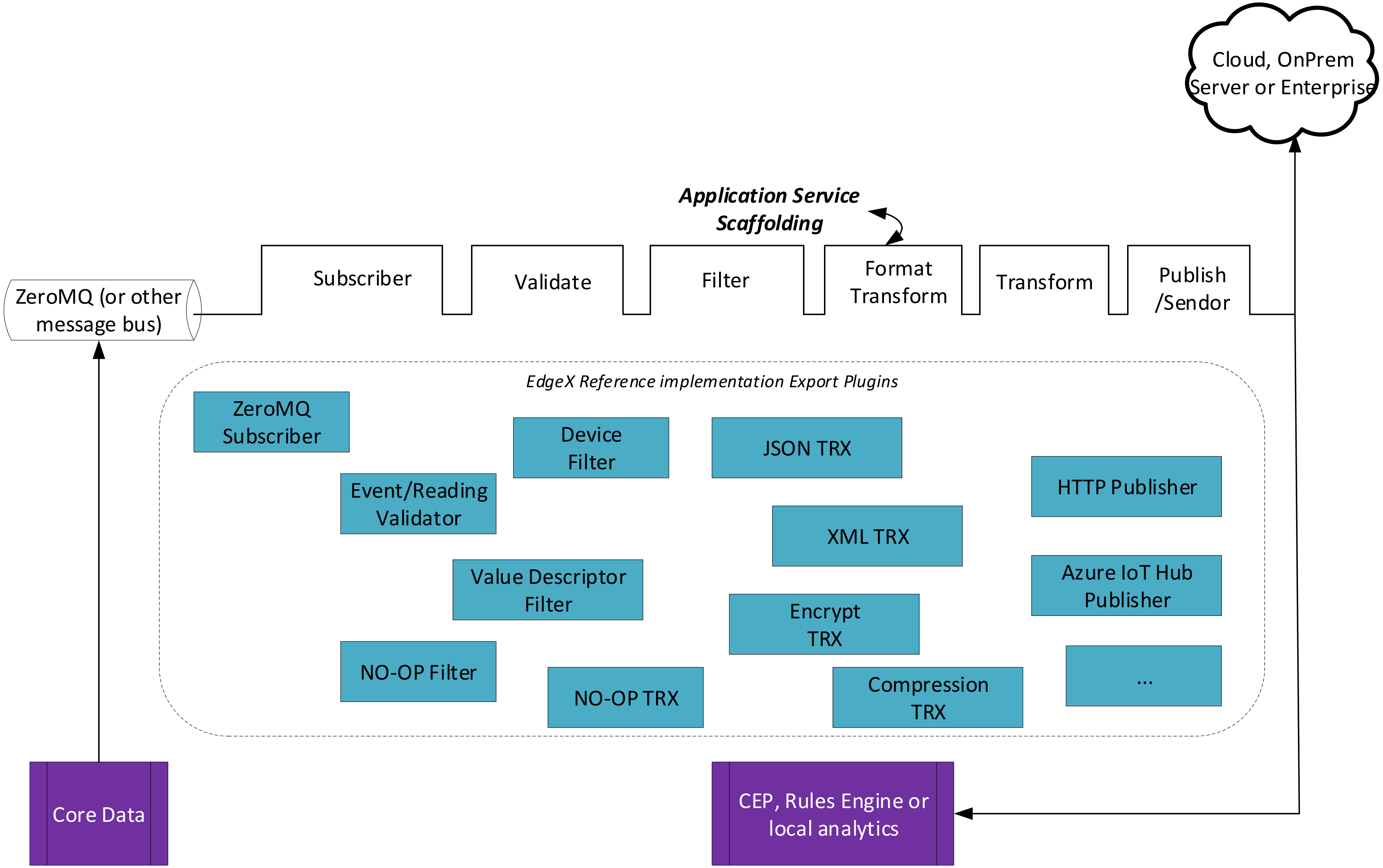
Implement by functions or by service?



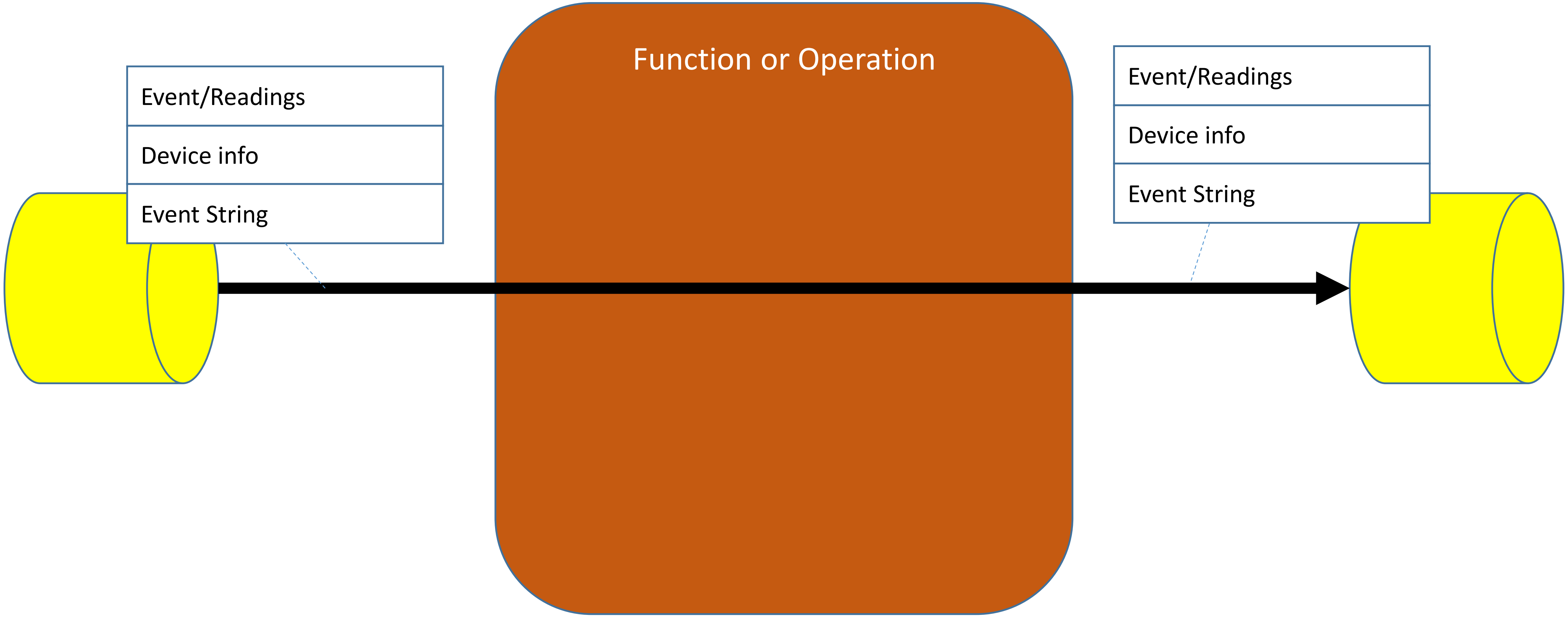
How to orchestrate per client?
How to secure?
Functions can be reused across clients

Internally secure
Internally orchestrated
Per client micro service
Duplicate code in each service

What provides the “scaffolding”?



Application Services Exchange


Event/Readings
Device info
Event String
Function or Operation
Event/Readings
Device info
Event String