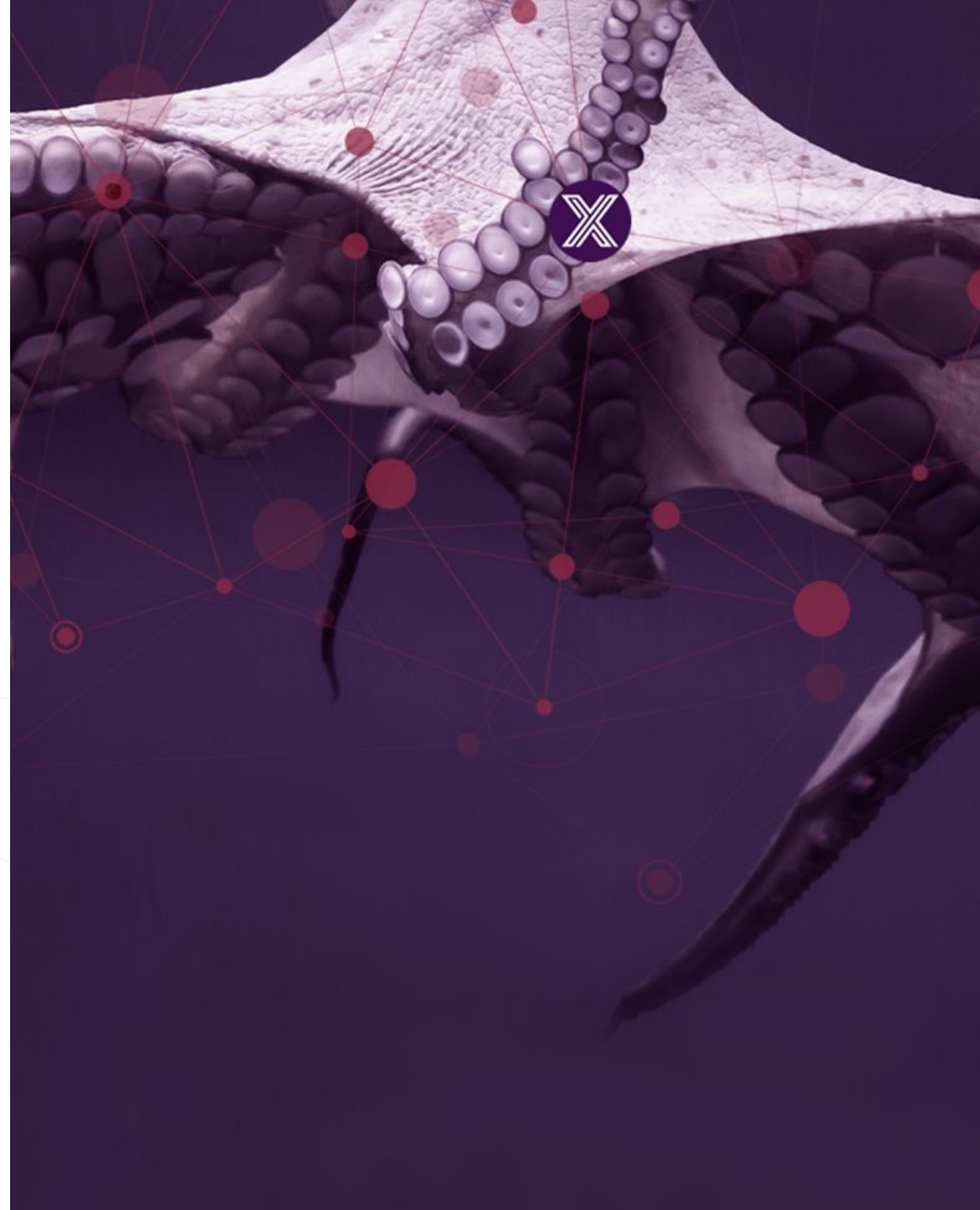




Application Services Design

Application Working Group
2018-10-16



LF Antitrust Policy Notice

- EdgeX Foundry meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at EdgeX Foundry meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Meeting Logistics

- Time: October 16, 2018 11am PDT – 12am PDT

Join from PC, Mac, Linux, iOS or Android: <https://zoom.us/j/611544838>

Or iPhone one-tap :

US: +16465588656,,611544838#

or +16699006833,,611544838#

Or Telephone:

Dial(for higher quality, dial a number based on your current location):

US: +1 646 558 8656 or +1 669 900 6833

or +1 855 880 1246 (Toll Free)

or +1 877 369 0926 (Toll Free)

Meeting ID: 611 544 838

International numbers available: <https://zoom.us/u/aoLL4E9yo>

Today's Agenda

- Internal microservice architecture



EDGE X FOUNDRY™

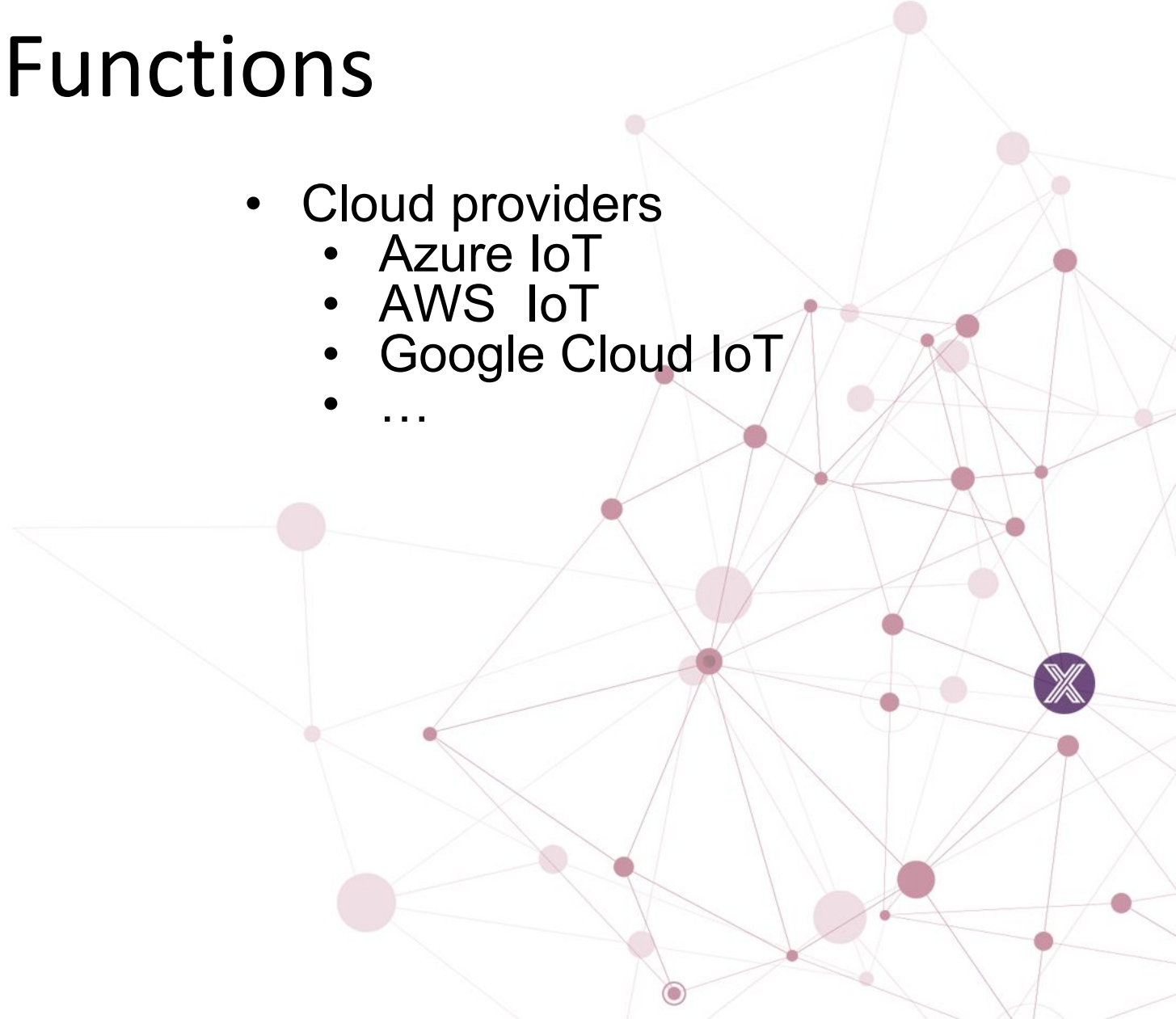
Application Service Functions

Application Services Functions

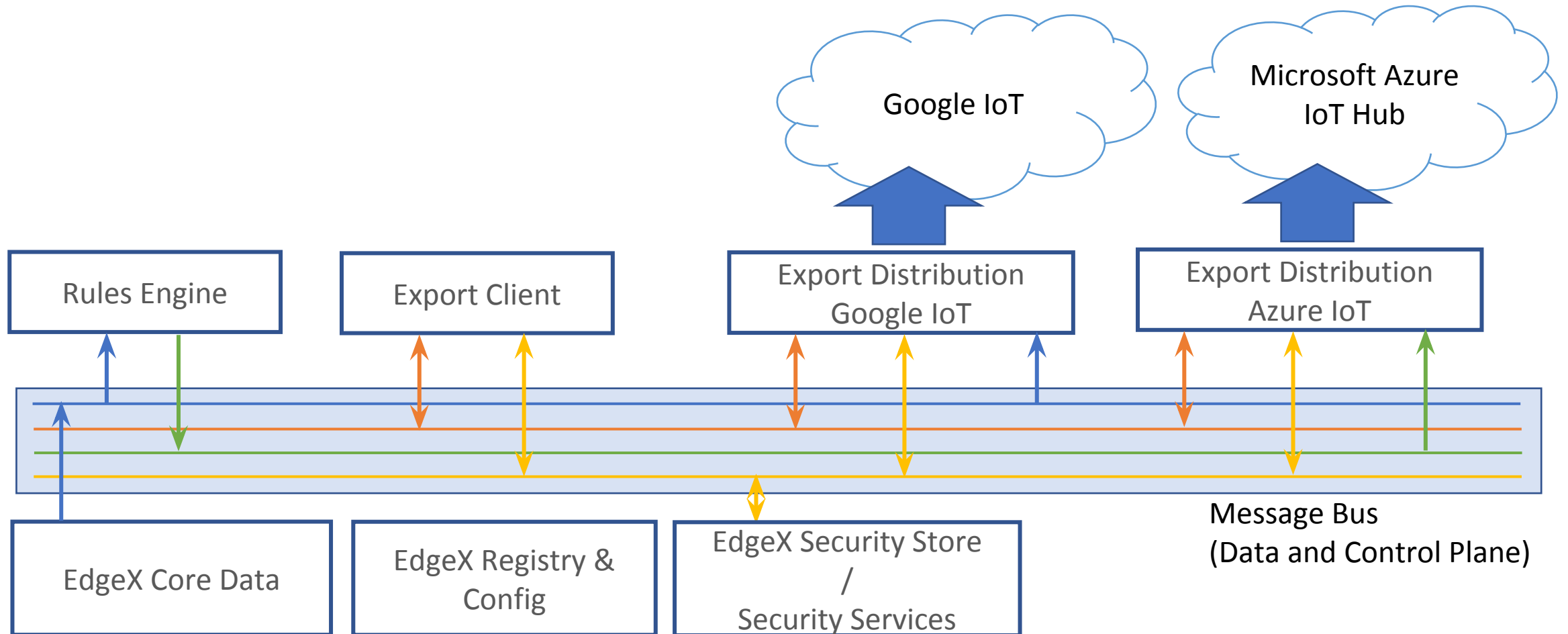
- Essentially an EAI engine
- Operations
 - Filter (only give me readings from device A; only give me readings regarding temperature, ...)
 - Validator (device ID, reading against value descriptor, ...)
 - Transformation (convert C to F values, convert CBOR to Protobuf, ...)
 - Enrich (add device metadata 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

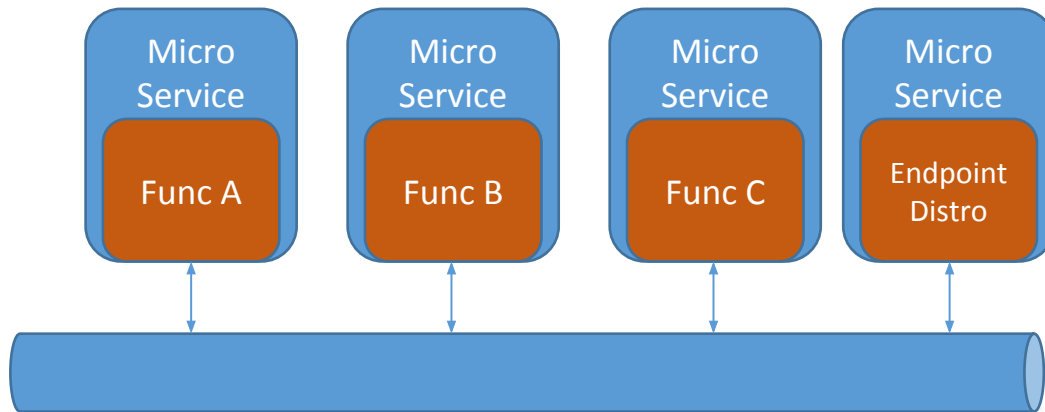
- Endpoints
 - HTTP(s)
 - MQTT(s)
 - AMQP
 - XMPP
 - WebSockets
 - CoAP
- Cloud providers
 - Azure IoT
 - AWS IoT
 - Google Cloud IoT
 - ...



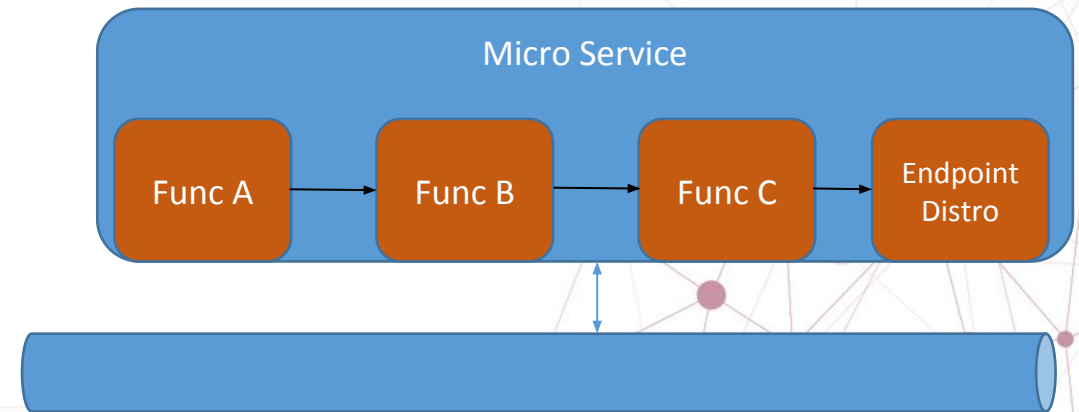
General Architecture Diagram – Multiple Export Targets



Implement by functions or by service?



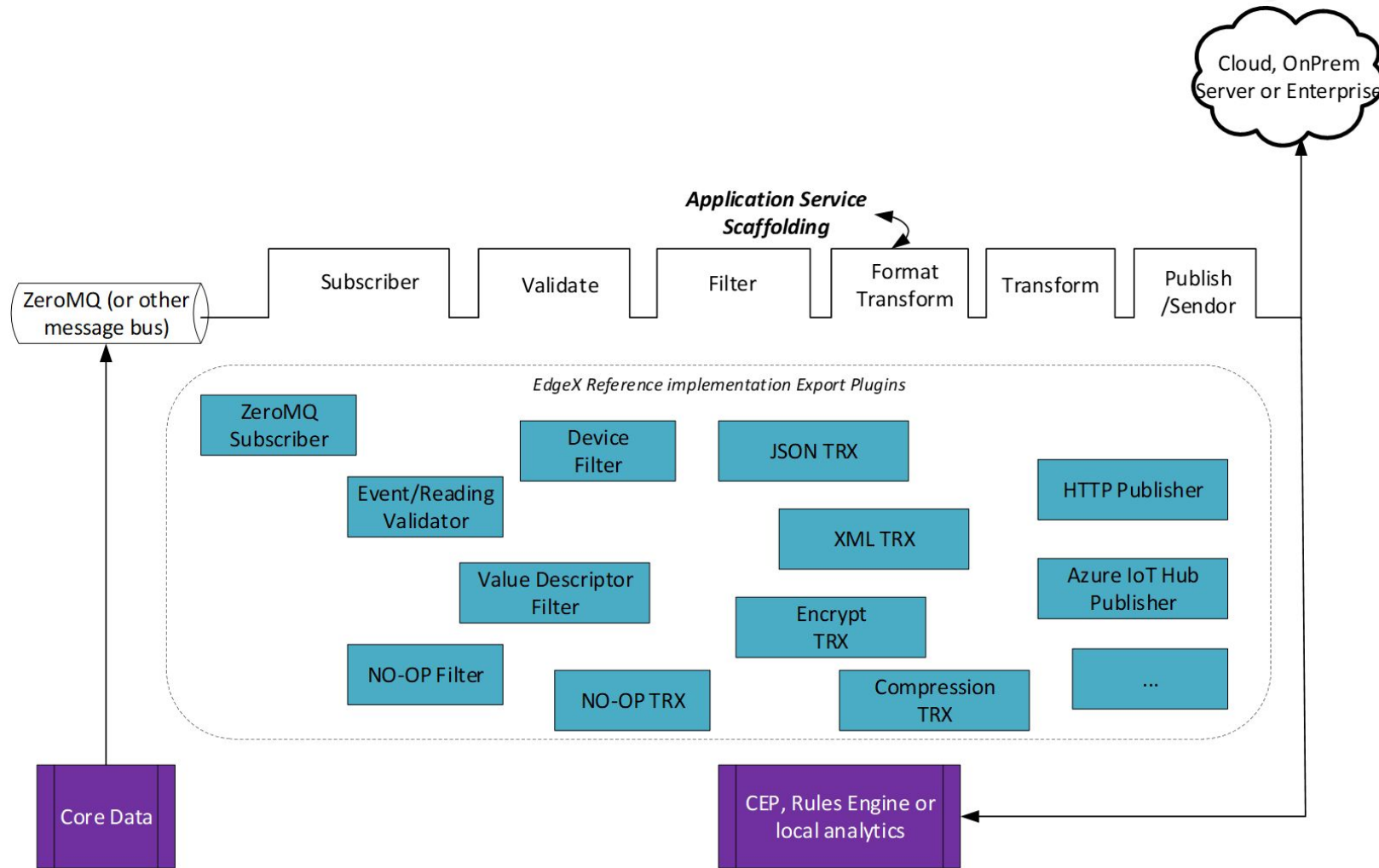
OR



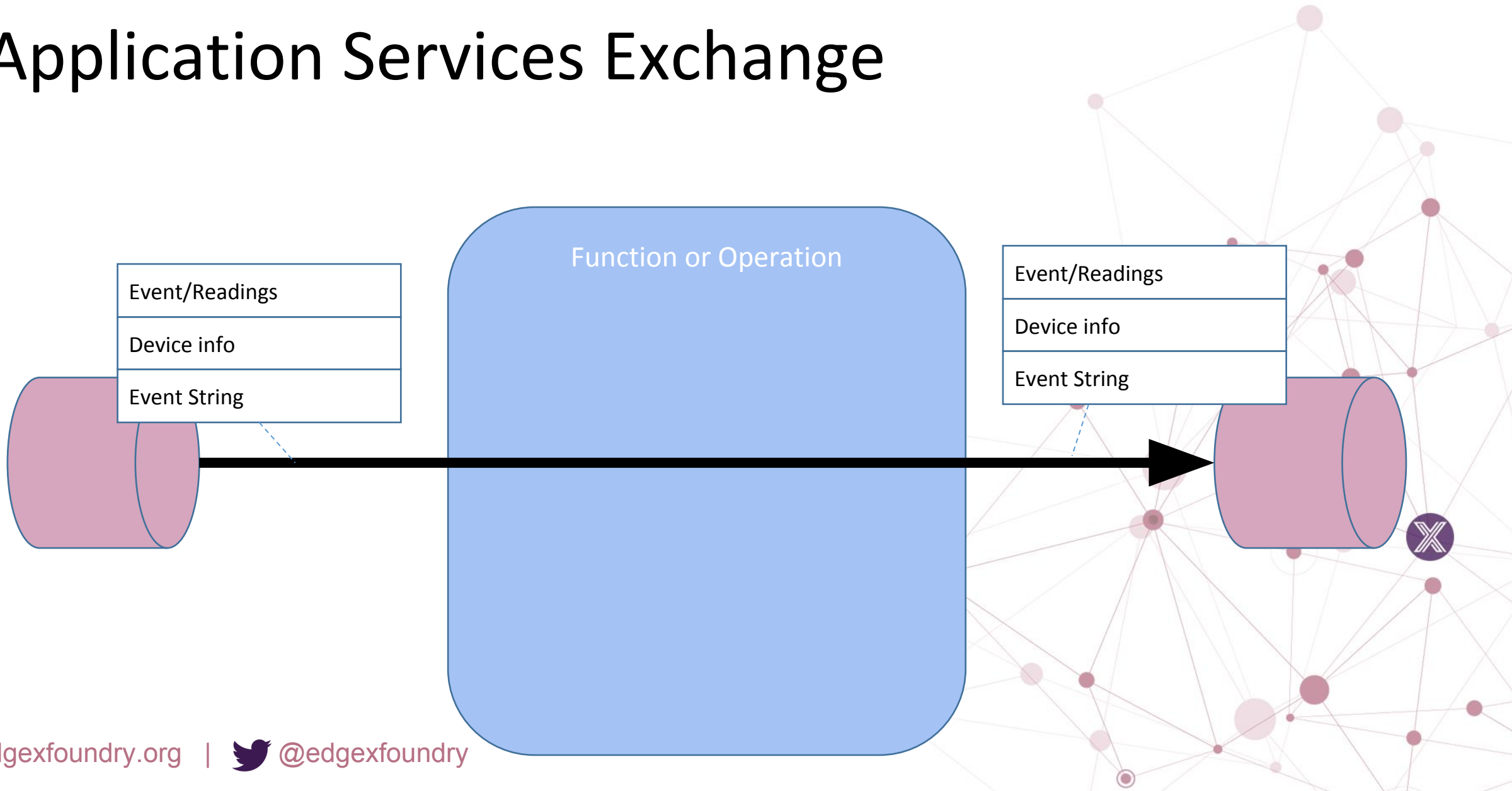
How to orchestrate per client?
How to secure?
Functions can be reused across clients

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

What provides the “scaffolding”?



Application Services Exchange



Internal Service Architecture

- Metrics and Tracing –
 - Do we need it?
 - Is the export of metrics and logs part of the System Management?
 - Do we need to expose endpoints for Kubernetes health-check
- Do we need to separate control plane from data plane for application services or it is (or will be) handled in System Management?
- Event Streaming / Event Sourcing:
 - Do we need to publish events on message bus?
 - Connected to export services xyz.
 - Export Service xzy1 configured...

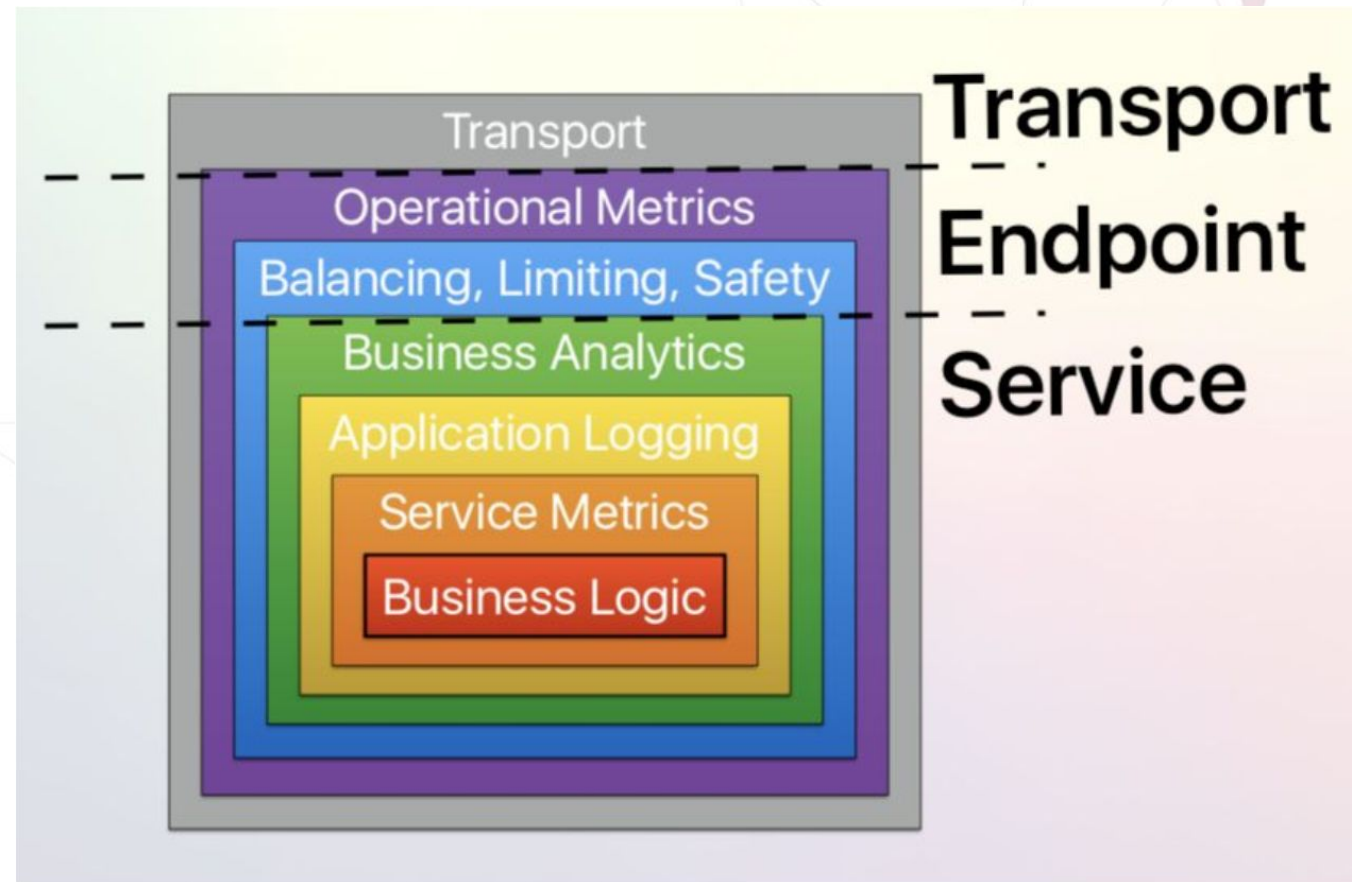
Using Go Kit - Benefits

- Go kit is a collection of Go (golang) packages (libraries) that help you build robust, reliable, maintainable microservices.
- You should use Go kit if you know you want to adopt the microservices pattern in your organization. Go kit will help you structure and build out your services, avoid common pitfalls, and write code that grows with grace.
- Go kit de-risks both Go and microservices by providing mature patterns and idioms, written and maintained by a large group of experienced contributors, and validated in production environments
- Go kit fills in the gaps left by the standard library - PC safety, system observability, infrastructure integration.

Go Kit Benefits - Onion Layers

1. Transport layer
2. Endpoint layer
3. Service layer

Requests enter the service at layer 1, flow down to layer 3, and responses take the reverse course.



Using Go Kit - Downside

- Too verbose: a. Function in the interface (make sense) b. Implementation (make sense) c. Endpoint factory function d. Transport function e. Request encoder, request decoder, response encoder and response decoder. f. Add the endpoint to the server g. Add the endpoint to the client.
- When using the go-kit, your endpoints get an `interface{}` object and return an `interface{}`, error tuple. You need to explicitly write the conversion to your implementation function. Actually, your endpoint factory will almost be a copy-paste of the following function:



EDGE X FOUNDRY™

Why FaaS is the Right Move

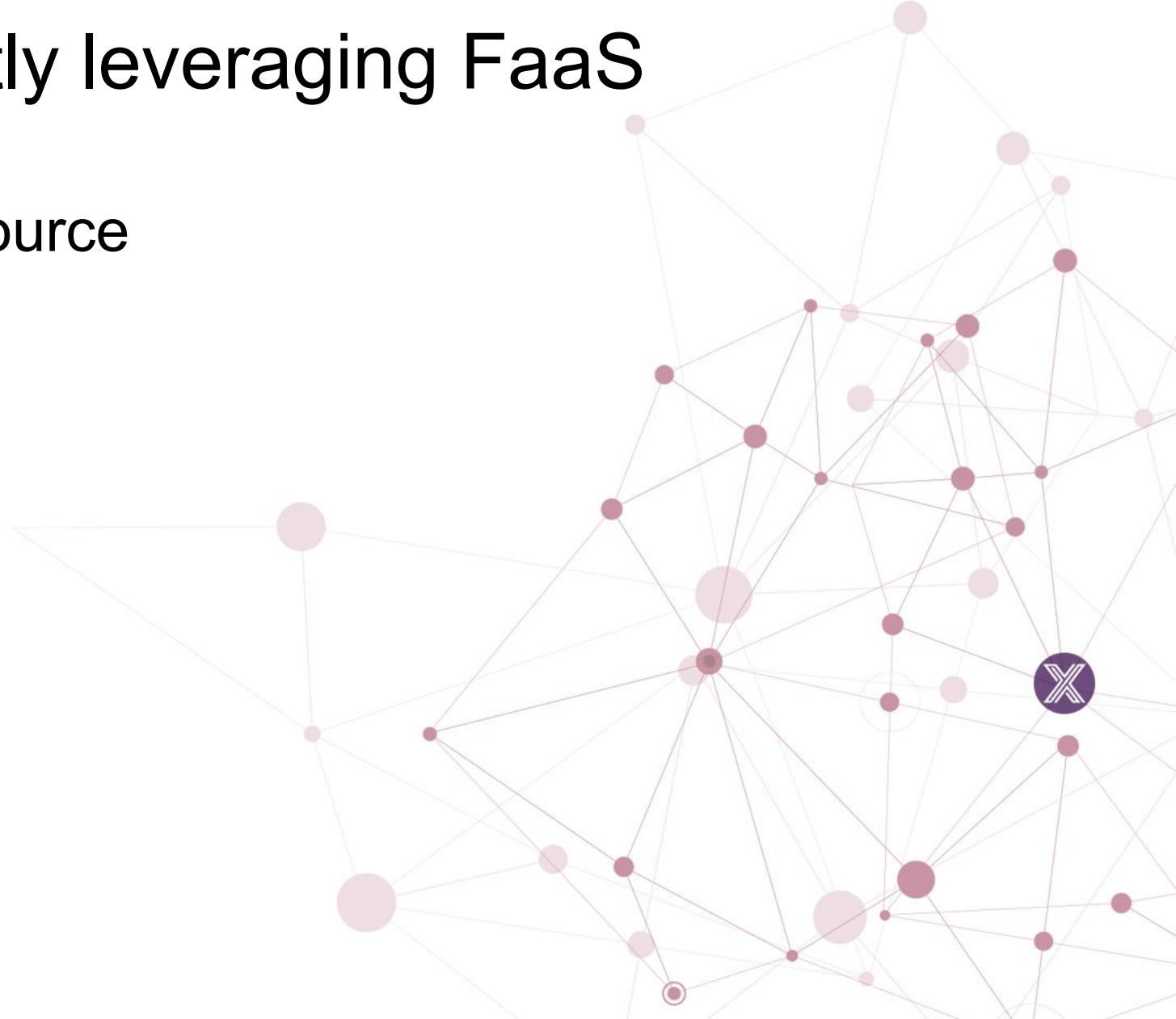
Michael Johanson (Intel), Lenny Godell (Intel)

Edge Platforms currently leveraging FaaS

- Azure IoT Edge – Open Source
- AWS Greengrass
- Target's UniMatrix

Coming Soon:

- EdgeX



Recommended order of FaaS projects

1. Nuclio – Easy to use
2. OpenFaas – Easy to use and get started
3. IronFunctions – API seemed to work, however UI wasn't so helpful.

Didn't Evaluate:

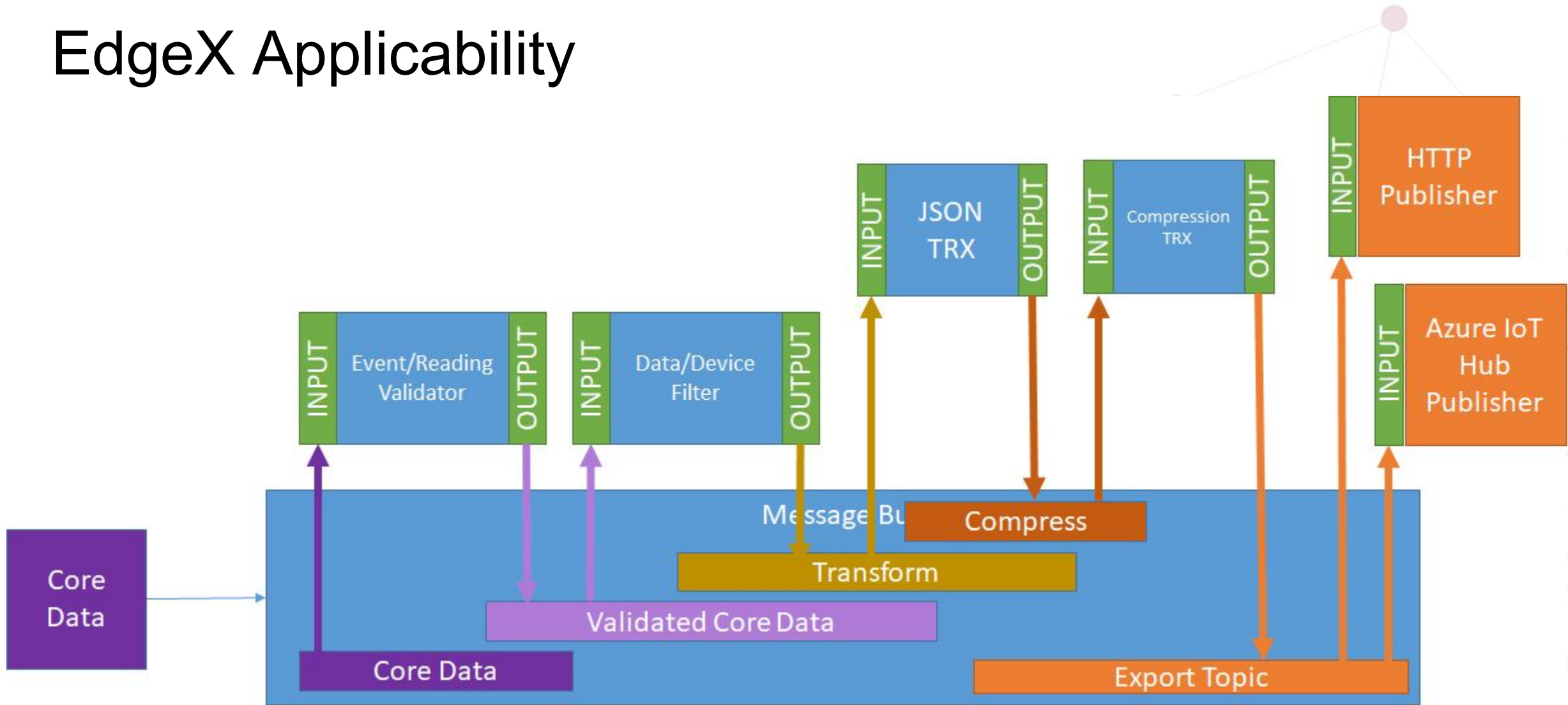
4. OpenWhisk didn't seem to have an easy way to get started
5. Kubeless – locked into Kubernetes

Top 3 In order recommendations

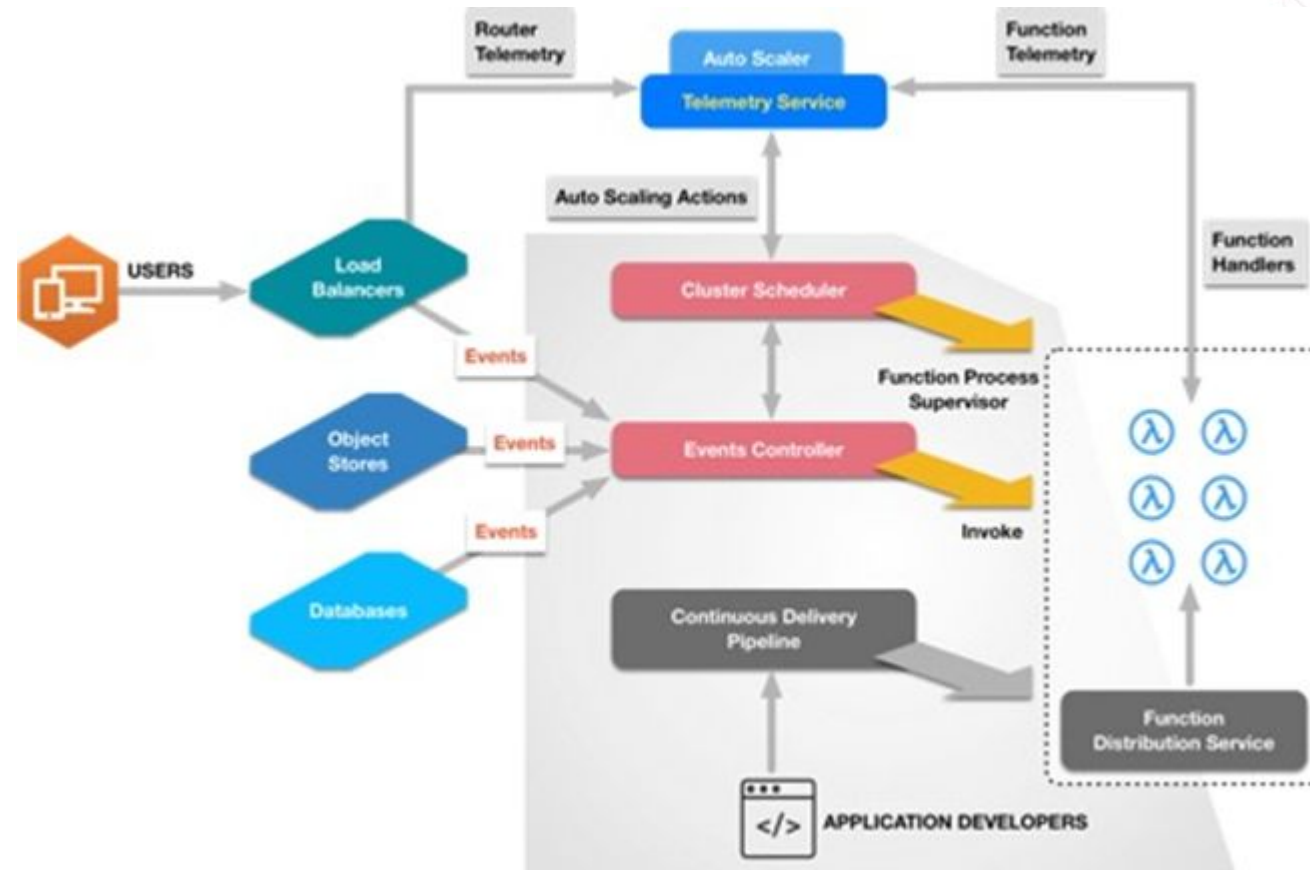
Framework/Platform	Docker Based	Language Support	Ease of use (1 Low -5 High)	Triggers	License
Nuclio	Y	Go, Python, .NET Core, Java, JS/NodeJS, Shell, Ruby *	5	Cron (Schedule based), EventHub, HTTP, Kafka, Kinesis, NATS, RabbitMQ, v3ioStream, CLI	Apache 2.0
OpenFaas	Y	*	3.5	HTTP (default) CLI	MIT
IronFunctions	Y	Go, JS, Ruby *	2	HTTP	Apache 2.0

Not Tested: Kubeless, OpenWhisk, AzureFunctions

EdgeX Applicability



Anatomy of a Serverless Architecture



“Serverless” adoption growing at rate faster than containers

- <https://stacksense.io/cloudopinion/expert-opinions/serverless-will-grow-at-a-faster-rate-than-cloud-did/>
- <https://read.acloud.guru/serverless-is-eating-the-stack-and-people-are-freaking-out-and-they-should-be-431a9e0db482>
- <https://www.infoworld.com/article/3265457/serverless-computing/why-serverless-is-the-better-option-than-containers.html>
- <https://blog.spotinst.com/2017/07/16/serverless-at-the-edge/>
- <https://thenewstack.io/week-numbers-serverless-adoption-par-containers/>
- <https://www.forbes.com/sites/forbestechcouncil/2018/05/18/why-companies-are-adopting-serverless-cloud-technology/#7c84737574d9>