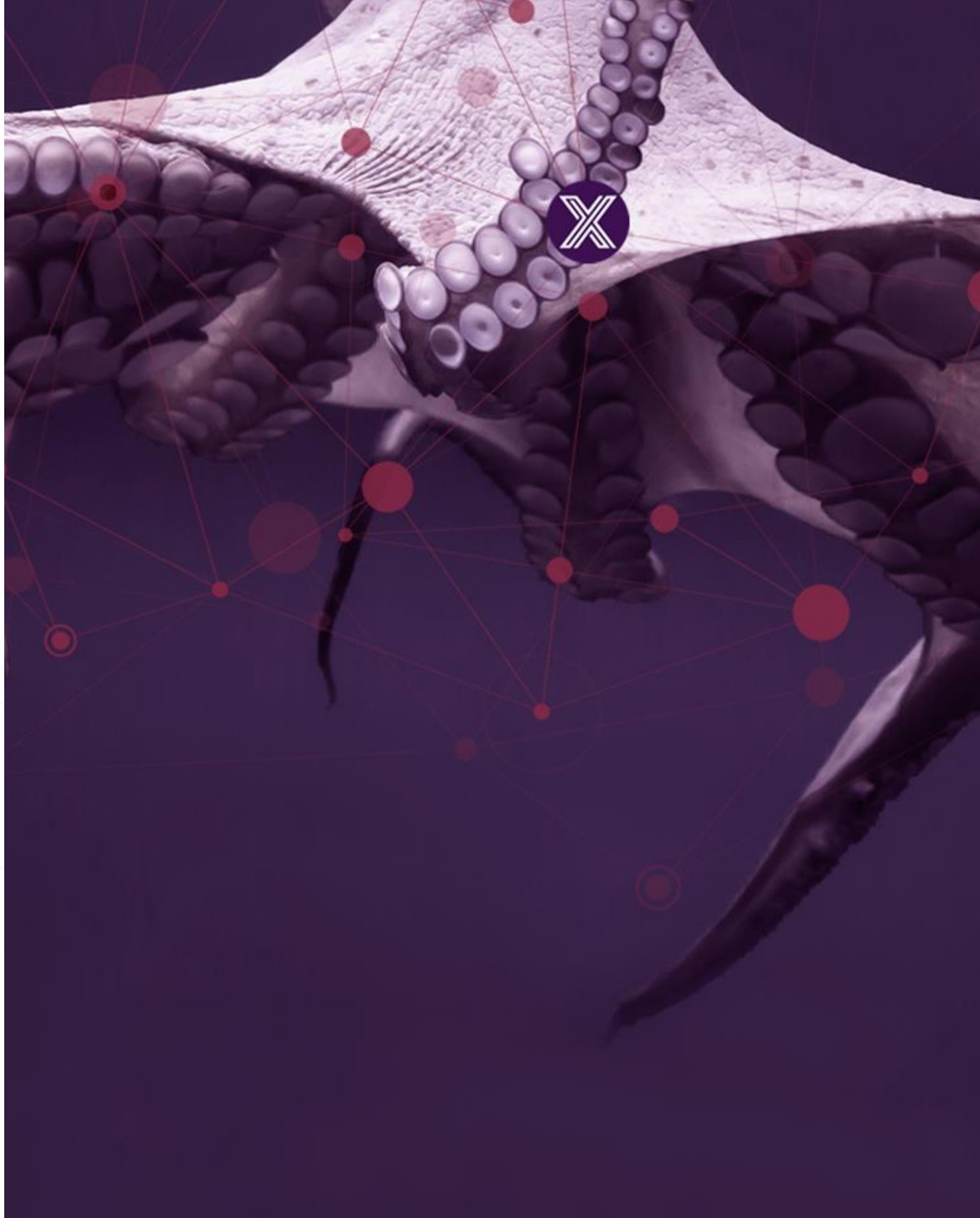
EDGE X FOUNDRY™

# Oil & Gas
# Business Cases

Alberto Dellabianca - October 2017



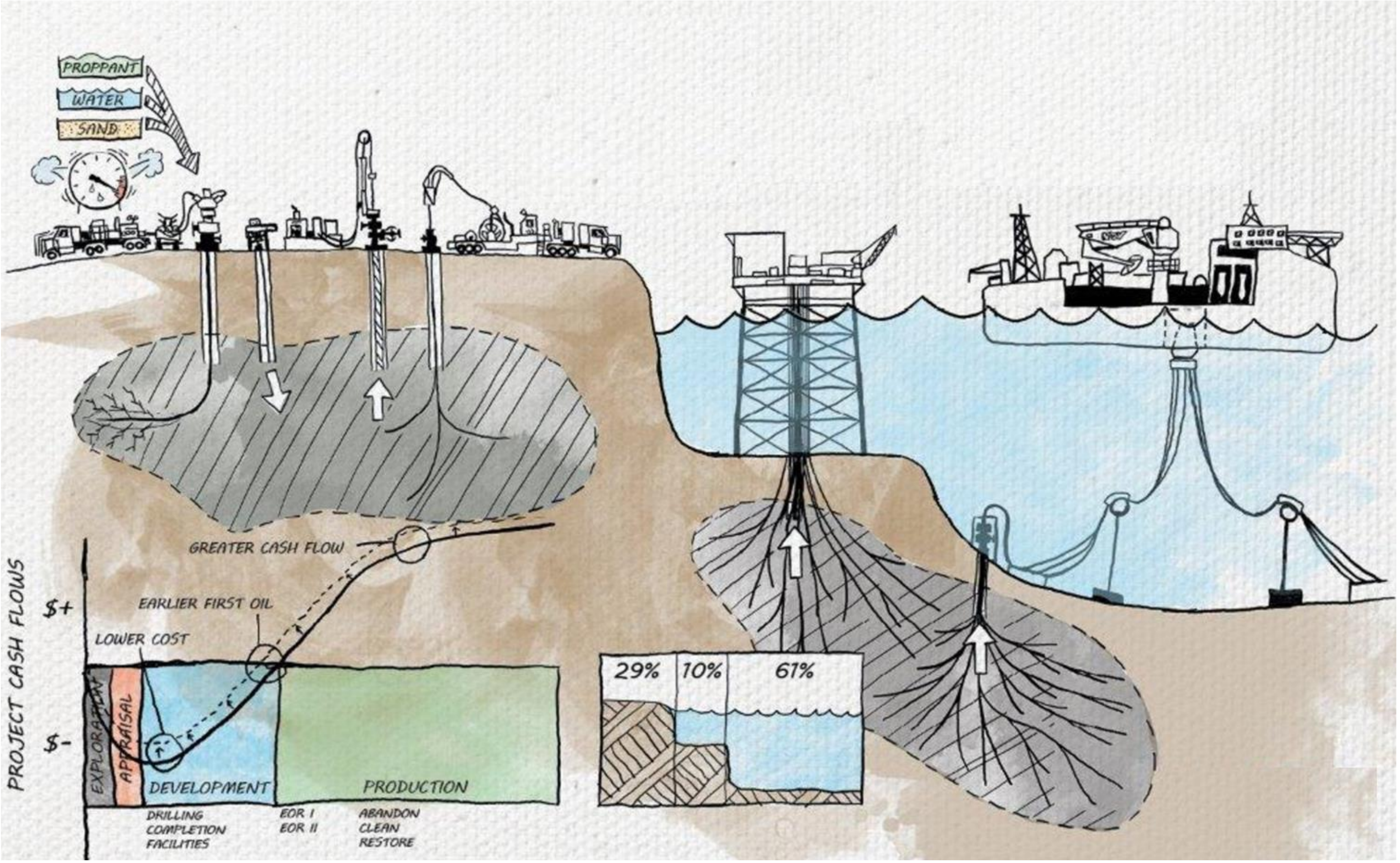
# Agenda

- Business Context
- Use Cases & Requirements
- Edge Vision
- Proposal for Next Steps

EDGE X FOUNDRY™

# Business Context

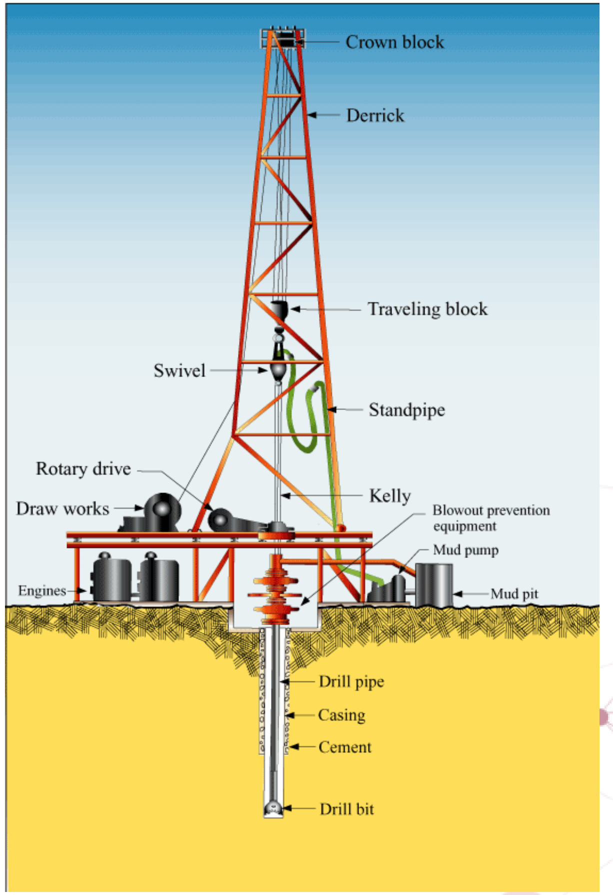
# Upstream Process

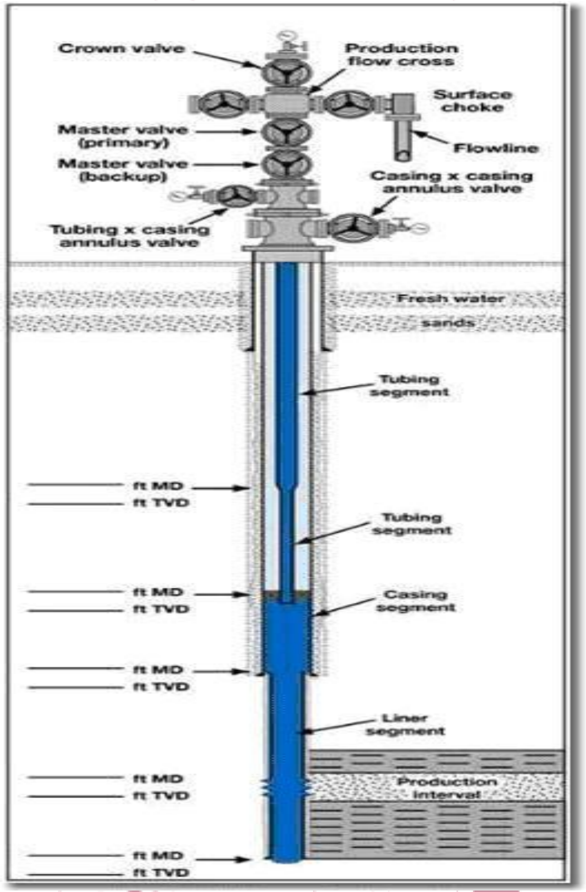



- Projects with long lifespan (20-30 yrs)
- Automation and sensors in all phases from drilling to production
- Traditional control systems and related architecture are dominant
- IIOT adoption is slower compared to other industries

# Onshore Drilling

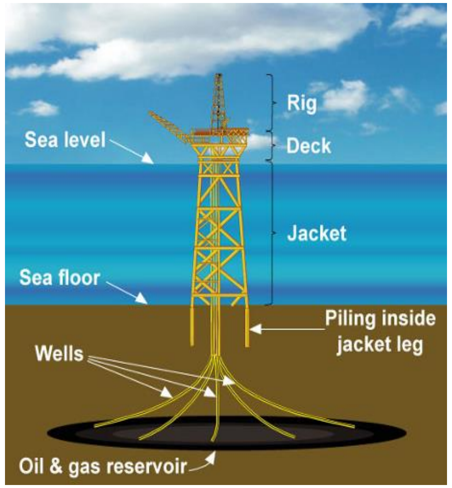
- Rig moving from one well pad to the other by road
- 1-2 months of activity for a single well
- Sensors, control systems and automation in place
- On-site optimization services as well as remote monitoring



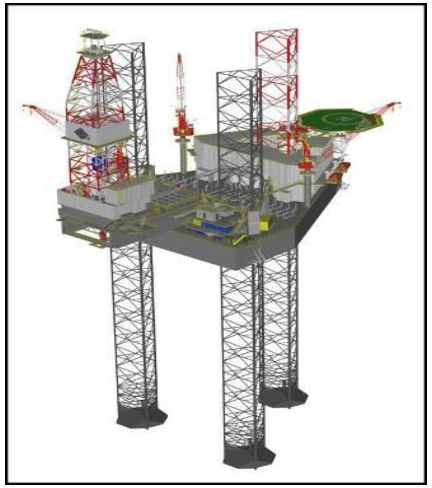



# Offshore Drilling

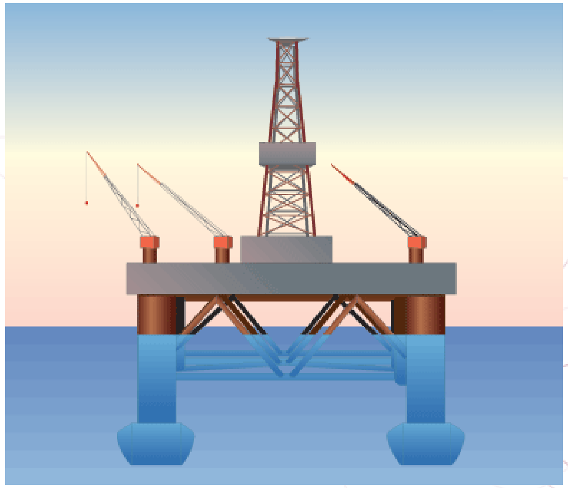
- 2-3 months of activity for a single well
- Advanced Sensors, control systems and automation with extensive safety features
- Different type of rigs based on water depth and other conditions




Fixed Rig



Jack-up



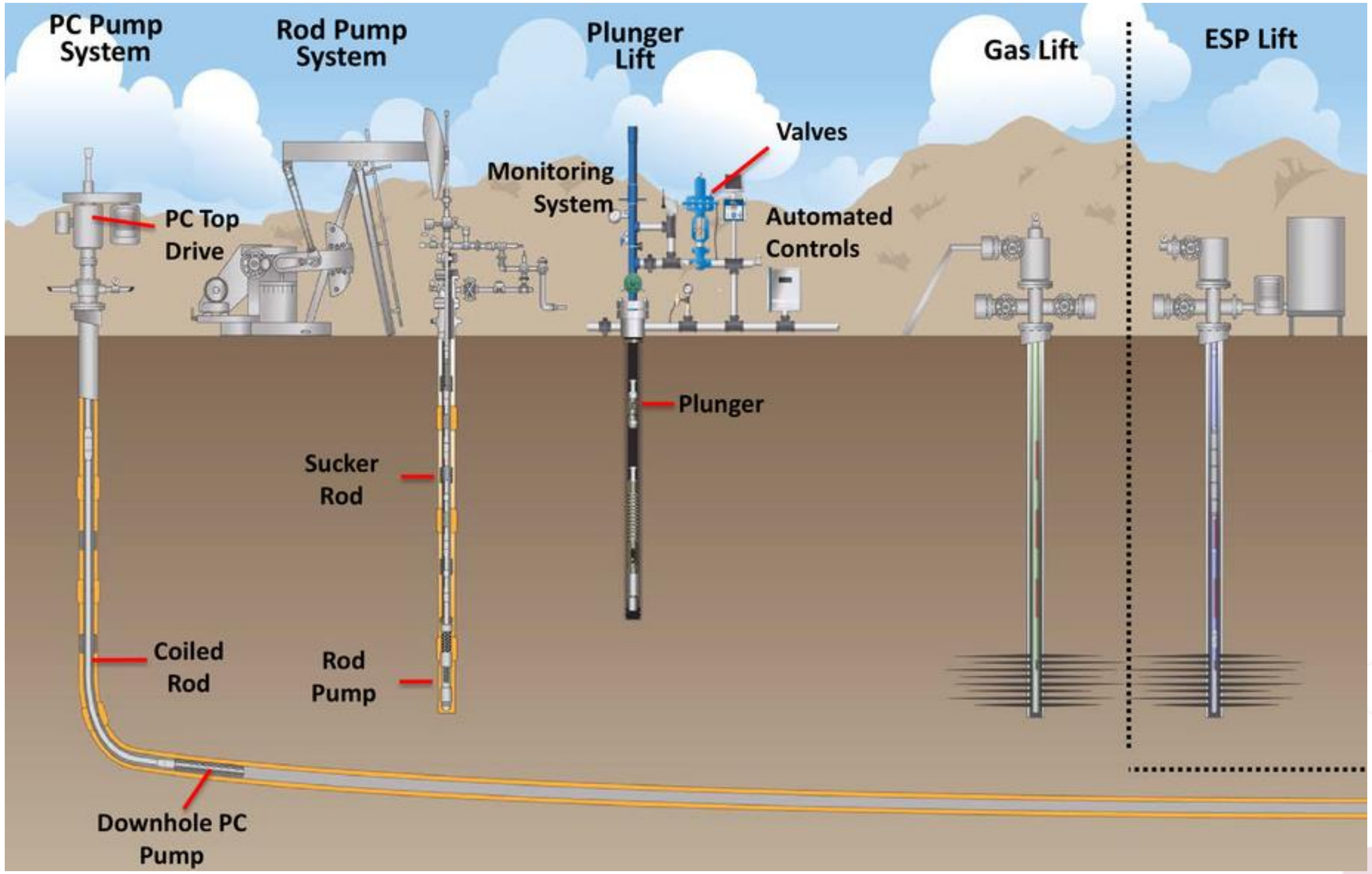
Semisub

Drilling Ship

# Well Intervention & Stimulation

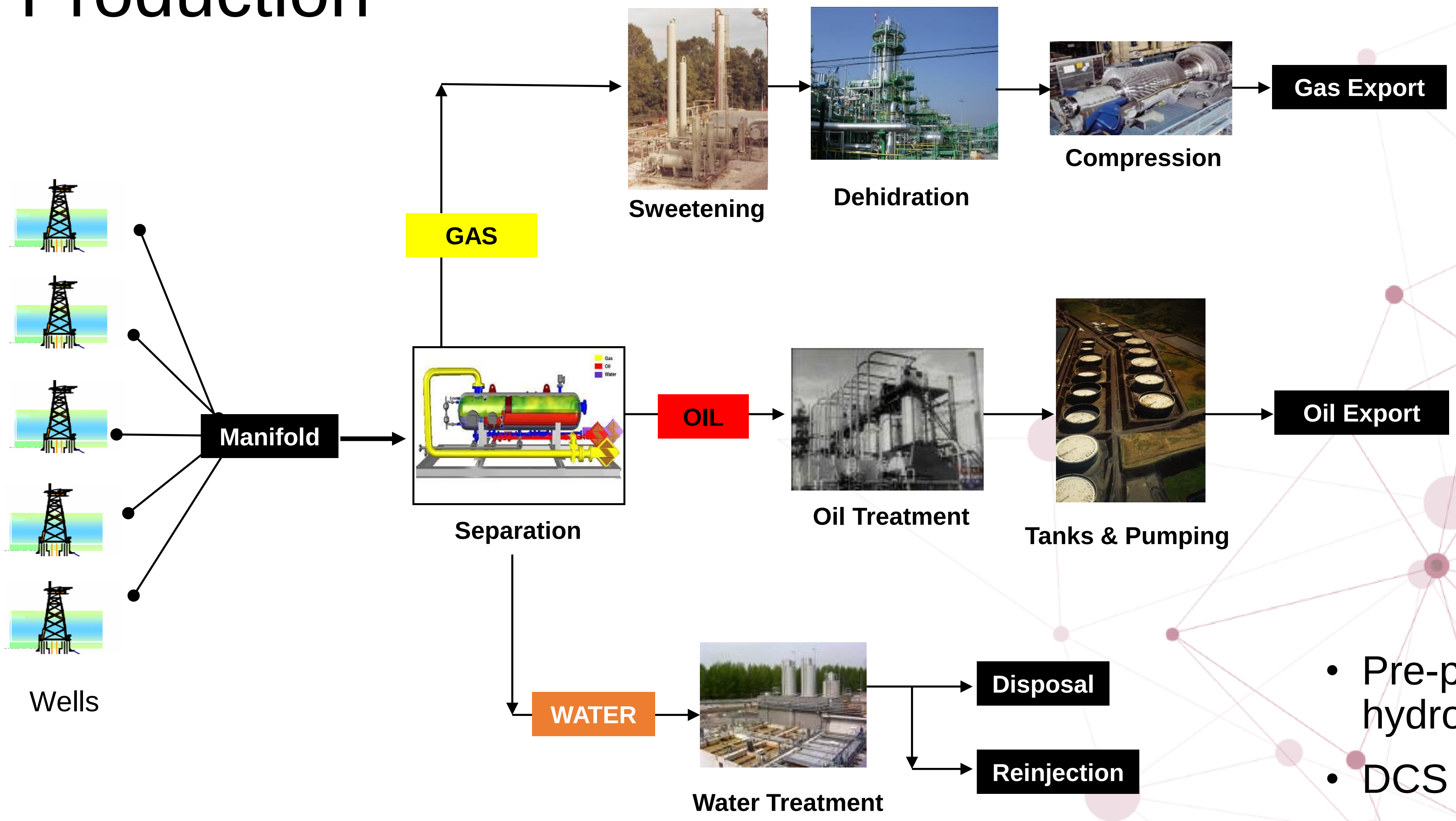
- Quick interventions (1-3 weeks per well)
- Each mobile unit with its own control system
- Frac Spreads for pressure pumping
- Coiled Tubing services
- Wireline services

# Artificial Lift




- System to help the well produce liquids
- Can stay onsite for the duration of the well (10-20 years)
- Surface and subsurface permanent components
- Local PLC in surface cabin

# Production



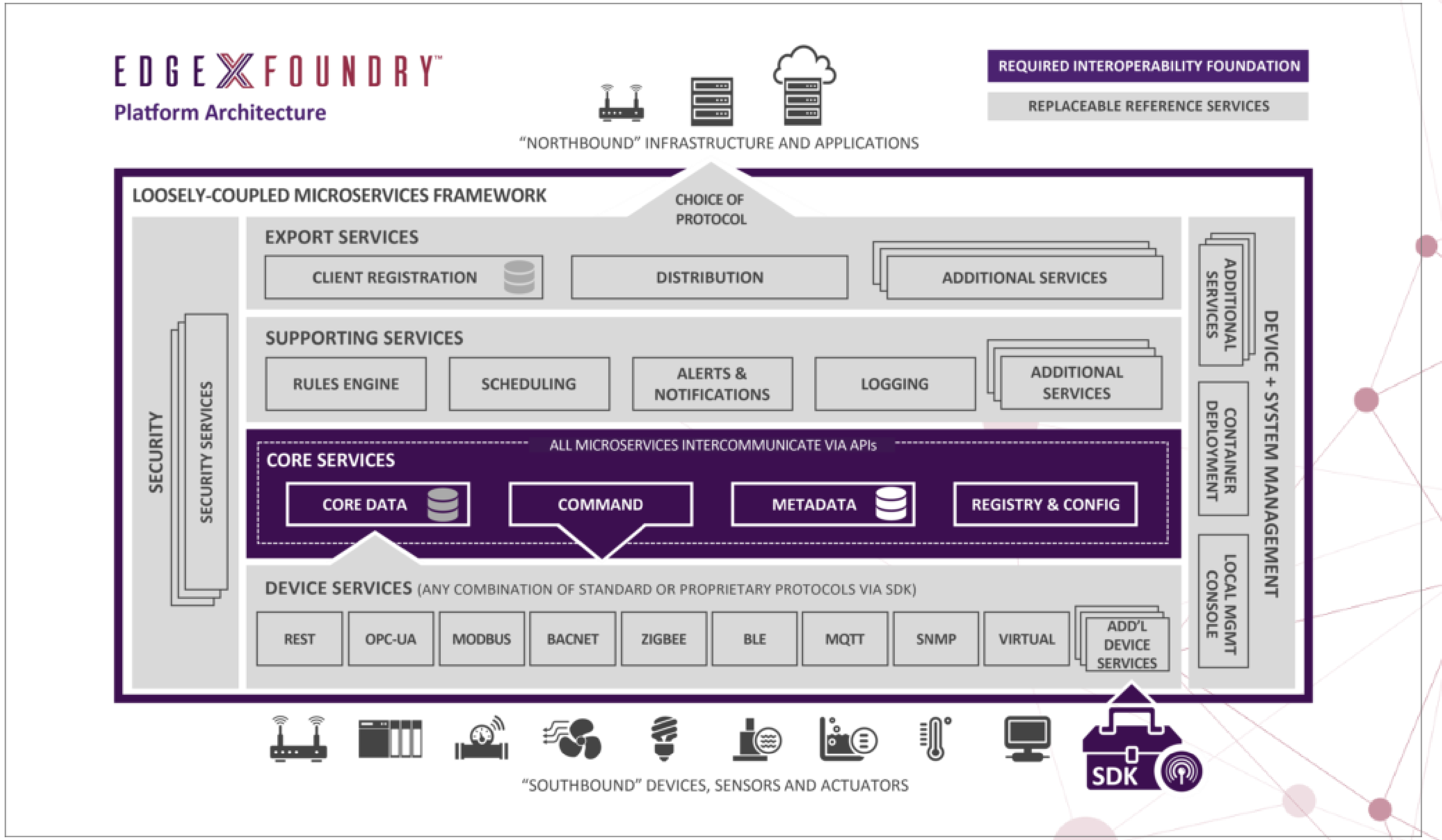

- Pre-processing of hydrocarbon
- DCS / SCADA for control

# Points of attention

- Fields are in remote areas, the software stack tends to become antiquate, difficult to upgrade and secure → remote manageability
- Challenging interoperability in the field between entities (Operators, Service Providers, Contractors) → standardization
- Business model not cost effective, redundant/overlapping Hardware and Software → microservices with plug-n-play architecture

# Use Cases & Requirements

# Current EdgeX – very flexible architecture



# Areas of attention

# Scalability

| | |
|---|---|
| **Standalone Equipment**<br>10-1000 channels | **Well**<br>200-1000 channels |
| **Onshore Facility**<br>3000-10000 channels | **Offshore Facility**<br>5000-50000 channels |
| **Onshore Rig**<br>1000-5000 channels | **Offshore Rig**<br>10000-100000 channels |
| **Well Intervention Unit**<br>50-300 channels | **Frac Spread**<br>5000-10000 channels |

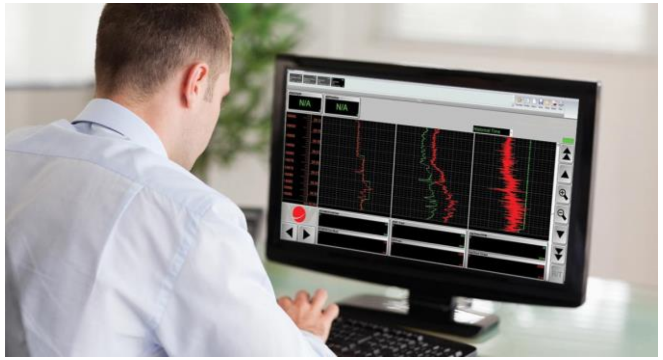
# Cybersecurity

- Oil & Gas not the most loved industry on our planet. Hacktivists targeted it publicly
- Nearly 70% of Oil & Gas companies were attacked in 2016
- Some of the risks faced in the case of a successful attack:
  - Plant shutdown
  - Equipment damage
  - Oil spills resulting in environmental damage
  - Safety measures violation resulting in injuries and death
- Geopolitics in the picture
- Without strong security foundations, EdgeX cannot even be considered

# Poor Connectivity

- Operating in remote zones with bandwidth constraints
- Satellite (very expensive) is the only option available in many cases
- Bandwidth assigned is often times low (e.g. 16kbps in North America to stream 1000 channels to the cloud)
- Local queuing mechanism to sync data as soon as the connection comes back after an interruption
- Possibility to backfill data in reverse order (most updated first)

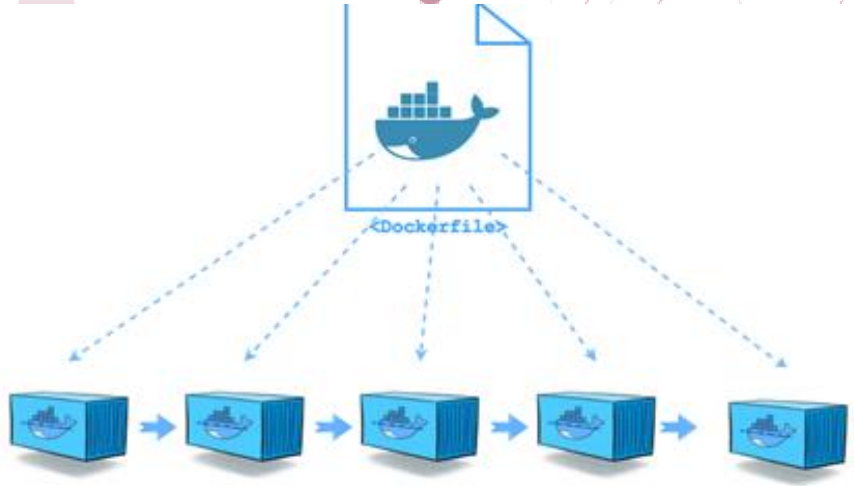
# Low Latency



- Many cases require a quick response from remote monitoring centers
- Latency < 3 sec is needed end-to-end (from data acquisition @edge to visualization on remote web portal). This means EdgeX latency from South to North shall be less than 500ms
- Direct streaming to the cloud bypassing Core Data?

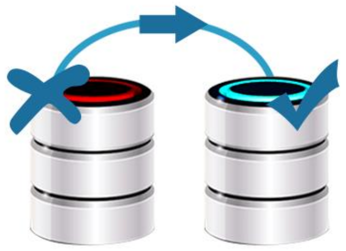
# Remote management

- Limited digital skills onsite
- Push remote updates on the configuration (update Core Metadata)
- Push updated docker containers for binary upgrades on one of the microservices
- Central web portal for all admin & remote management activities
- Kubernetes?



# Reliability

- Downtimes may have big safety and economic consequences
- High Availability architecture (East-West connections) with 2 nodes in failover
- Scheduled (e.g. every night) backup of all Core-Metadata config to the cloud to enable an easy swap of hardware in case of replacement with a new provisioned EdgeX



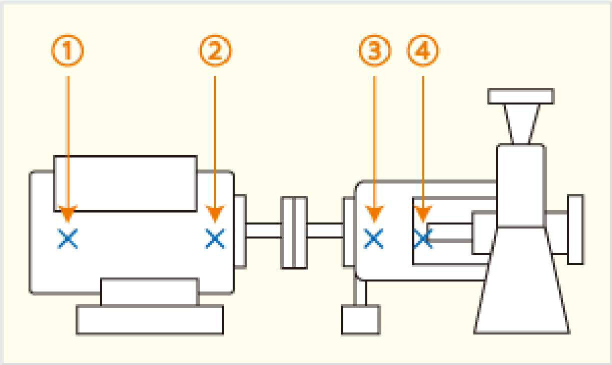
# Local Sharing

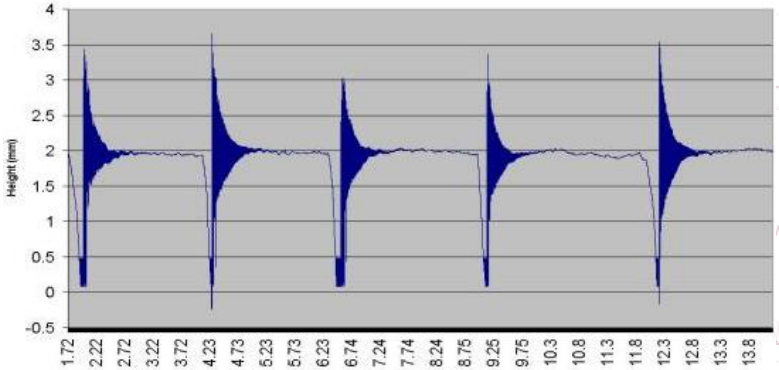
- Local OPC UA server exposing Core Data and Core Metadata
- This is necessary to share information to non-edgeX devices
- Interoperability between different entities

OPC UA™

# High-Frequency

- Vibration Sensors with data at 10khz
- High speed protocols such as DDS are used. EdgeX Device Service for Open DDS?
- Why not using a Time Series Database as Influxdb within Core Data? That would provide native time-series functions used for local apps and analytics (Influxdb is written in Go, runs well in docker, supports batch writes and has a REST API to interact with)



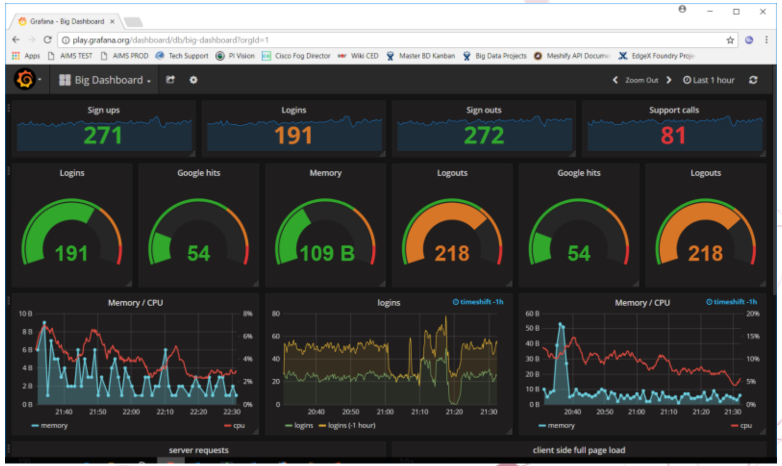



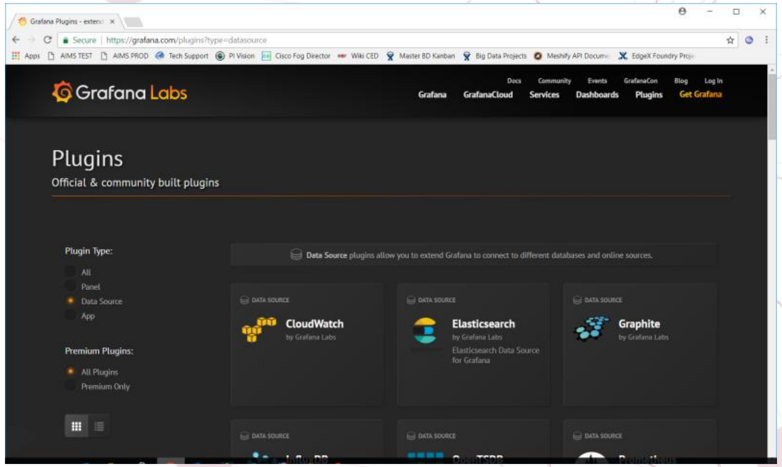
# Analytics

- An offshore rig generates between 1TB and 2TB data per day
- It’s impossible to send all data to the cloud → edge analytics
- R and Scikit learn Python engines may be too heavy to run on an edge device
- Possibility to save the model as PMML (Predictive Model Markup Language) which is a de facto standard
- Have a microservice to run a lightweight (Go? C/C++?) machine learning engine accepting the PMML model as config input as well as the reference to core data inputs to feed the model

# Visualization

- Local visualization solution is required
- Grafana is a perfect match (open source, built in Go, runs on docker, web- based, light footprint, very fast, lots of features, extensible with plugins)
- Would need a Grafana “Datasource plugin”
- Where to save config data?



# Legacy Systems

- OPC DA is a first class citizen now and for years to come
- Running OPC DA on Linux is challenging… New device service using OpenOPC which leverages on pyro?
- Modbus Serial still present in many cases

# Edge Vision

# Ideas for mid-long term

- Asset Template
- App Store
- East/West Interop.
- Industrial Sharing
- AI Agents

# Asset Template

- Goal: declare a specific instance of EdgeX to be an instance of a specific asset template (e.g. Pump model A1287B) and be able to push the relevant set of microservices and configuration

- The EdgeX asset template defines:
  - Which microservices are needed
  - Default configuration for microservices

- The central EdgeX Console should have a catalog of templates and be able to create instances based on this template

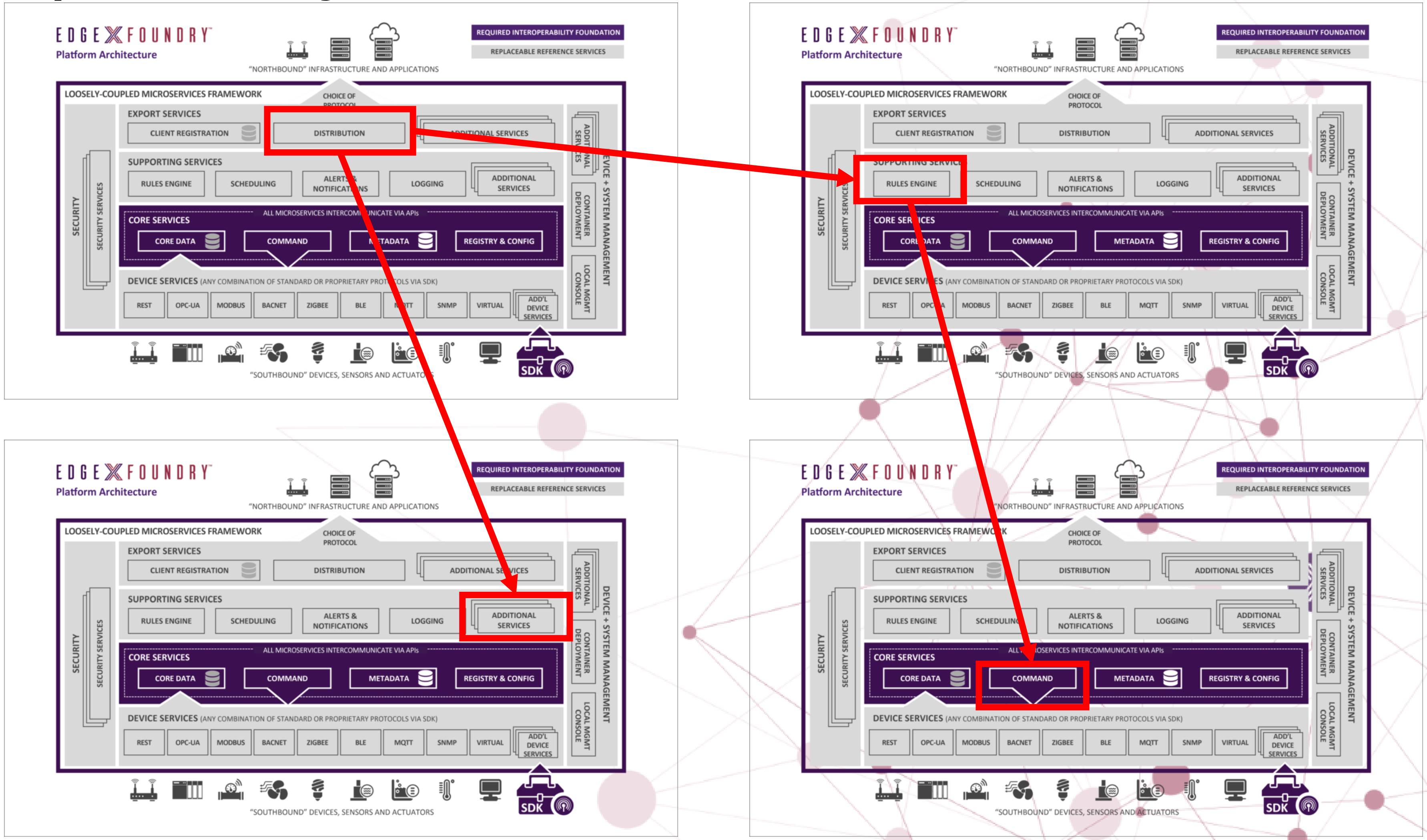
- Be able to upgrade or push the same changes to all EdgeX devices sharing the same asset template

# App Store

- EdgeX to become for Edge devices the same as “Android” for smartphones
- Commercial apps built by service providers and available on app store hosted by the project (an app could be the artificial intelligence model to optimize the asset)
- Business model which stimulates commercial players to contribute to the ecosystem (similar to GE Predix)

# East/West Interoperability

- Microservices able to invoke other microservices running on a different EdgeX instance
- Shall this be done through a broker?
- This could be useful for different service providers interacting in the field on behalf of the operator

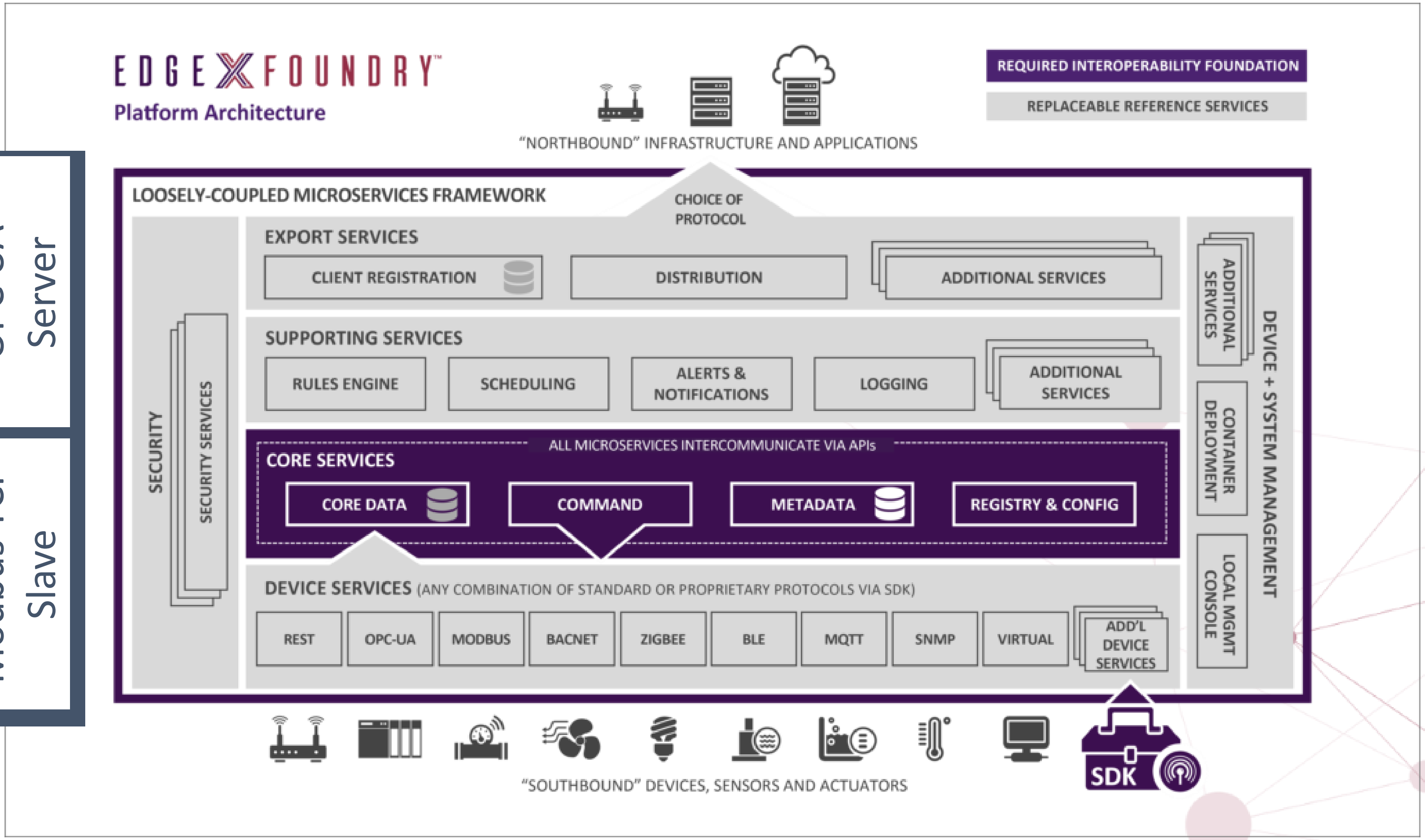


# Industrial Sharing with non-EdgeX devices

Industrial Sharing

OPC UA Server

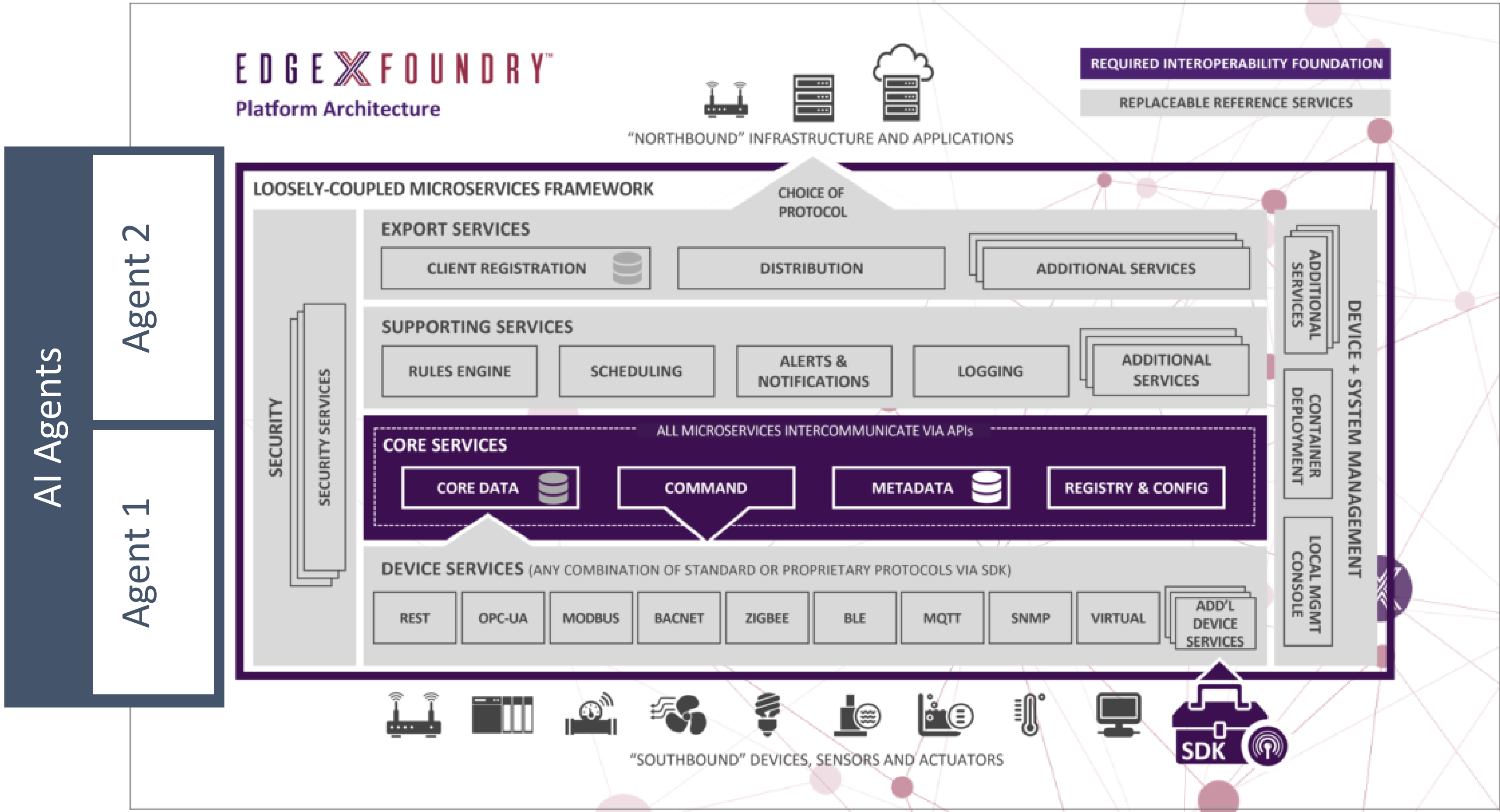
Modbus TCP Slave




- OPC UA Server and Modbus TCP Slave to expose data to other non-EdgeX devices

# AI Agents

- This would be the ultimate step to achieve full data-driven automated optimization
- AI Agents on different EdgeX devices talking to each others and negotiating the best course of action based on the overall process



EDGE X FOUNDRY™

# Proposal for TSC

# Proposal for TSC

- Project Lead - Alberto Dellabianca (National Oilwell Varco)
- Other Company to join this project - OEM, Service Providers, Operators. Few contacts so far, no official committment
- Deliver Specific Market Requirements to Other WGs – Presentation posted on wiki is a generic starting point. Specific requirements will be contextualized in EdgeX perspective (scorecard for each category / microservice)
- Create Reference Architecture for Specified Vertical / Use Case - Define 3 reference implementations: Single Asset, Onshore Facility, Offshore Facility
- Develop Unique Source Code Outside of Core - Osisoft PI Export, Native TSDB – Target California Release?
- Liaison with Appropriate SDOs and Consortia - Society Petroleum Engineers, Universities

# Specific collaborations with other WGs

**Security Services**

Provide requirements

**Export Services**

Export Service to Osi PI
Export Service to influxdb

**Support Services**

Test streaming analytics in high throughput scenarios

**Core Services**

Native TSDB collaboration?

**Device Services**

Provide requirements on which industry protocols are missing
Test high throughput scenarios with OPC UA

**System Services**

Provide requirements

Collaborate on mockups for a local management console

Define use case and workflow for backups to cloud

# Reference Architecture

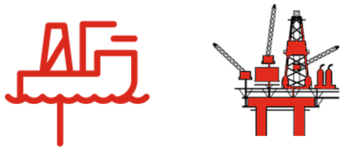
- Specify a reference architecture for 3 Macro Scenarios
- Recommended specs (minimum hardware & docker-compose)
- Test beds

Single Asset (Fixed/Mobile)

Up to 1,000 inputs/s

Onshore Facility

Up to 10,000 inputs/s



Offshore Facility

Up to 100,000 inputs/s

# Why OSI PI Export Service?

- Enterprise historian used by majority of operators
- The platform itself comes from the refining industry and later on expanded in Upstream

# Why a native TSDB?

- Massive amount of sensor and control data (ingestion volume and velocity)
- Need a local efficient persistent storage for long periods (offline scenarios)
- Need for native time series functions for local analytics

- Solution: local influxdb (Open Source, written in Go)

- 2 Options
  - Northbound Export service to influx as if it was an external microservice
  - Modify the Go version of “Core Data” in order to support as a backend influx (alternative to Mongo)

EDGEXFOUNDRY™

# Thank you!