



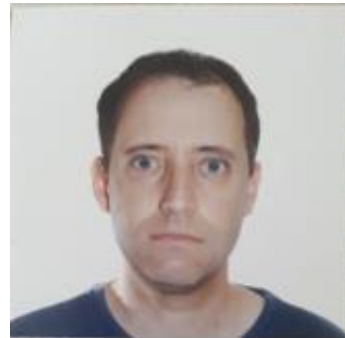
Using Argo and Knative to Orchestrate Media-

Intensive Services in 5G Edge

David Breitgand, IBM Research – Haifa
Cloud and Data Technologies,
Senior Technical Staff Member,
Research Scientist



Avi Weit, IBM Research – Haifa,
Cloud and Data Technologies,
Research Scientist



Pavel Kravchenko, IBM Research – Haifa,
Cloud and Data Technologies,
Developer



Outline



- Using Argo and Knative to Orchestrate Media-Intensive Services in 5G Edge
- Problem?
- Media-Intensive?
- 5G Edge?
- Orchestrate?
- Why Argo and Knative?
- How did we used them?
- Demo
- Conclusions



5G MEDIA



5G Media: Programmable edge-to-cloud virtualization fabric for the 5G Media industry <http://www.5gmedia.eu/>

5G-MEDIA delivers an integrated programmable service platform seamlessly spanning cloud and edge for development, design and operations of media applications in 5G networks. 5G-MEDIA emphasizes

- (a) user QoE,
- (b) fast development cycle,
- (c) scalability, and
- (d) cost-efficiency.



Telefónica Investigación y Desarrollo



Problem



- Facilitate deployment of a 3rd party software at the 5G edge
- Cost-efficiently
- Instantaneously elastic, but easy to scale down
- Fast time to market



Why is it difficult with VMs?

- We don't know in advance where and when to deploy VNFs
- We cannot keep VMs running everywhere waiting for events
- We cannot size VMs exactly
- We don't know how to create instantaneous autoscaling
- We cannot scale down a VM based auto-scaling group easily
- We cannot support bill per actual usage
- Many idiosyncrasies related to infrastructure

Serverless to the rescue!



- For **session-based, event-driven** workloads, such as:
 - Tele-immersive gaming
 - Distant learning
 - Holographic teleconferencing/telepresence
 - Mobile journalism
- Instantiate CNFs comprising a network service **just in time**
- **Where they are needed**
- For the **exact duration** of the time they are needed
- **Portable**
- Fast time to market, **no infrastructure issues** involved

Motivating Use Cases



Use Case 1

Tele-Immersive Media



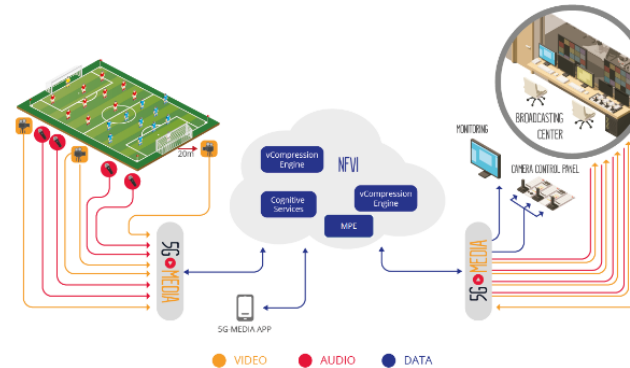
Goal: Ensure Quality of Experience for real-time multi-party applications, enabling HQ 3D virtual reconstructions of users



Main Expected Benefits: Improved QoE for players/spectators and support of real time Tele-Immersive applications

Use Case 2

Smart + Remote Media Production



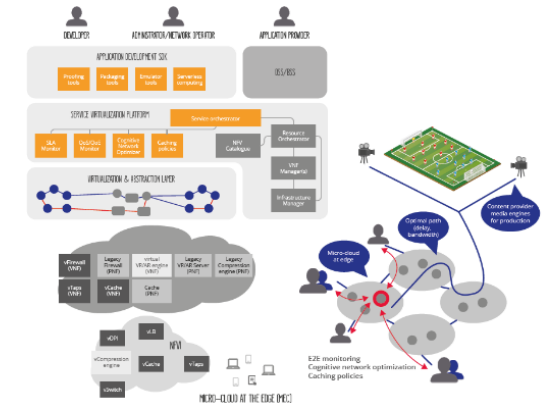
Goal: Provide broadcasters with ad-hoc, scalable, flexible and time-saving production mechanisms leveraging professional and user-generated remote media content



Main Expected Benefits: Reduction in costs, personnel, time and complexity for remote production, enabling exploitation of user-generated media content

Use Case 3

UHD over Content Delivery Network

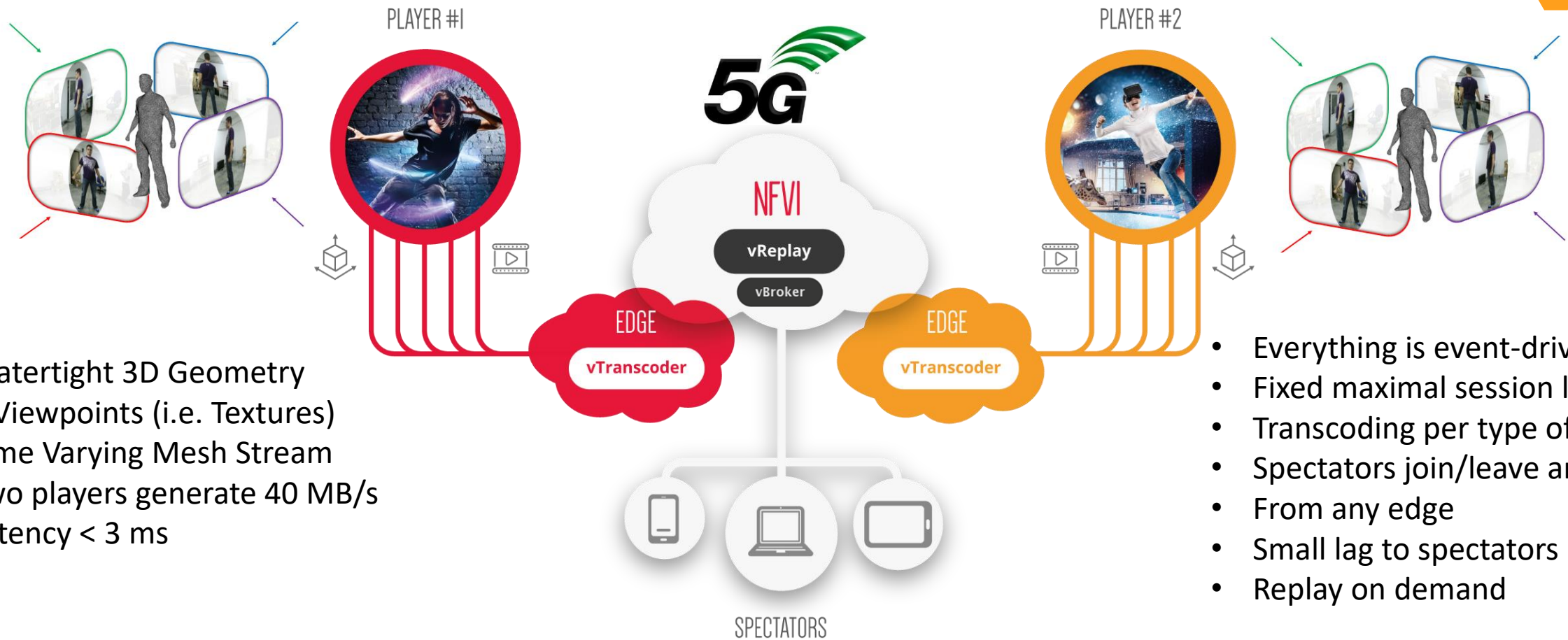


Goal: Deliver new capabilities to media service providers by distributing UHD content (4K and 8K) with an optimal consumption of resources



Main Expected Benefits: Better experience for end users and new market opportunities in content delivery

Tele-Immersive Media



- Watertight 3D Geometry
- 4 Viewpoints (i.e. Textures)
- Time Varying Mesh Stream
- Two players generate 40 MB/s
- Latency < 3 ms

- Everything is event-driven
- Fixed maximal session lifetime
- Transcoding per type of terminal
- Spectators join/leave any time
- From any edge
- Small lag to spectators is OK
- Replay on demand

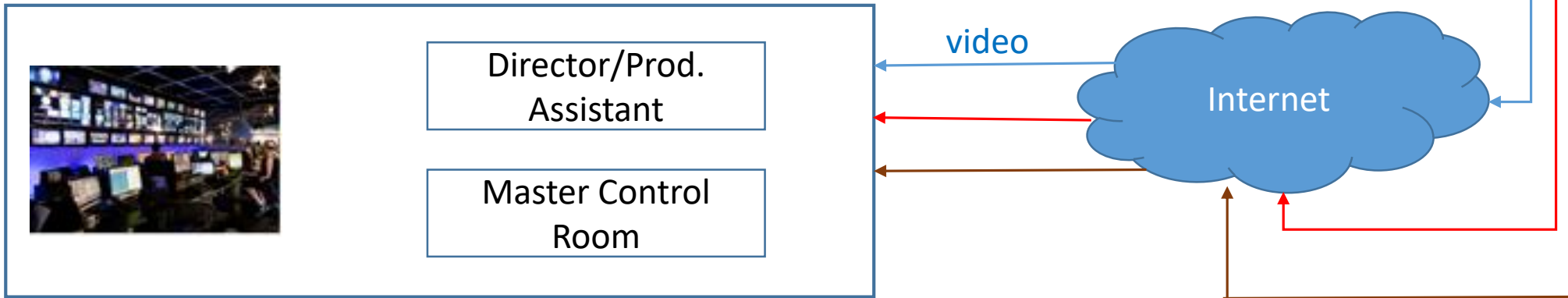
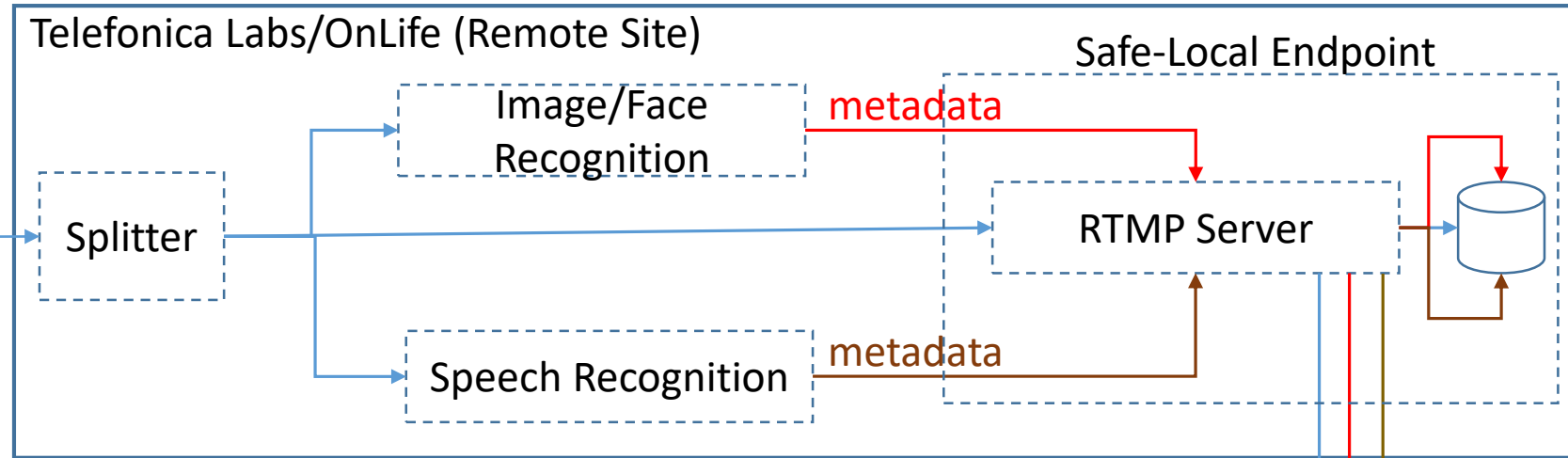
- Collaboratively with 5G-MEDIA Service Virtualization Platform (SVP), the application selects a most suitable edge for players
- With serverless, all functions are instantiated when and where needed, for the exact time they are needed
- Functions are dynamically orchestrated



Mobile Journalism – CNF Structure



5G



- Collaboratively with 5G-MEDIA SVP, the application selects a most suitable edge for a contribution
- With serverless, production support functions are instantiated when and where needed, for the exact time they are needed
- Production support services are dynamically orchestrated

Conceptually, 5G-MEDIA Followed ETSI MEC

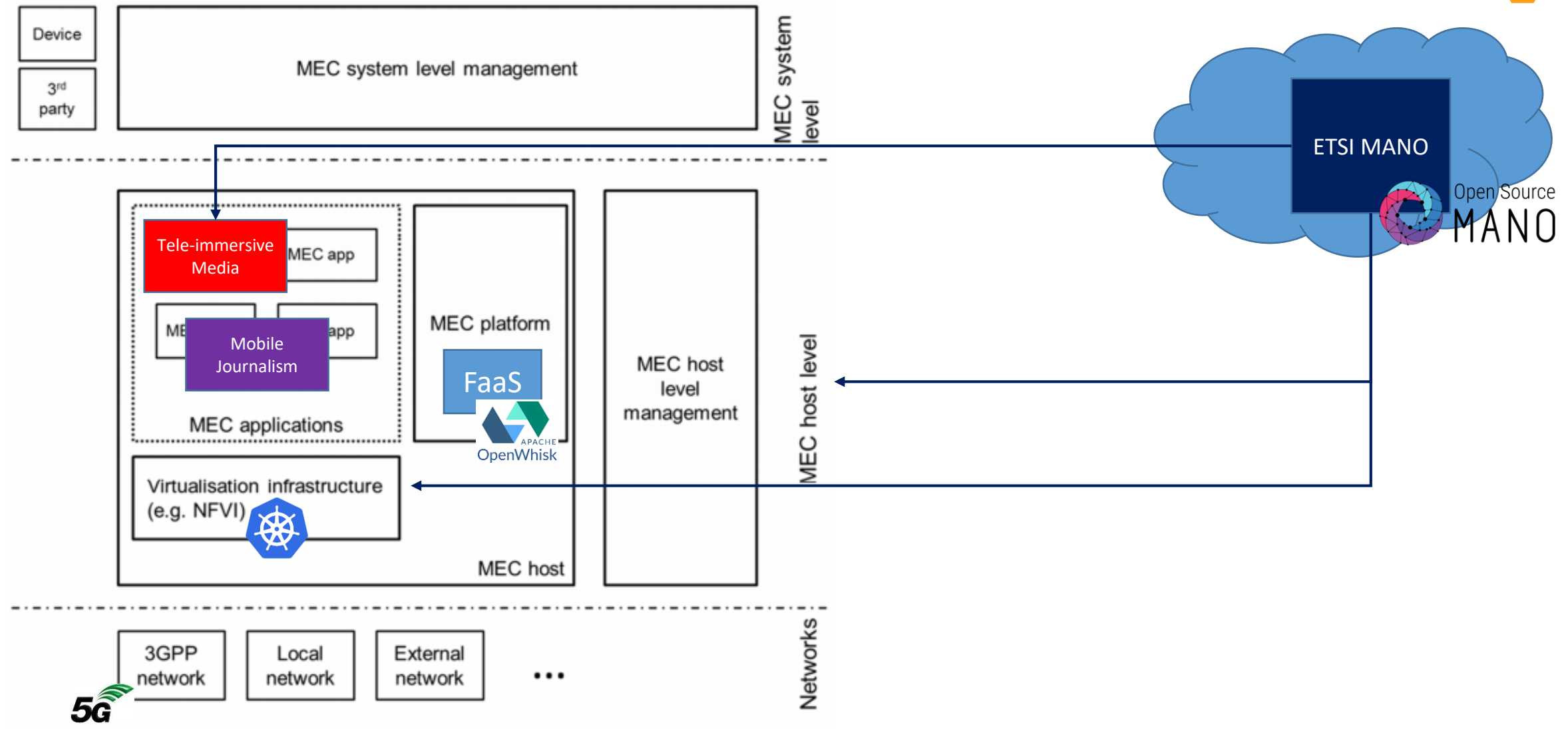
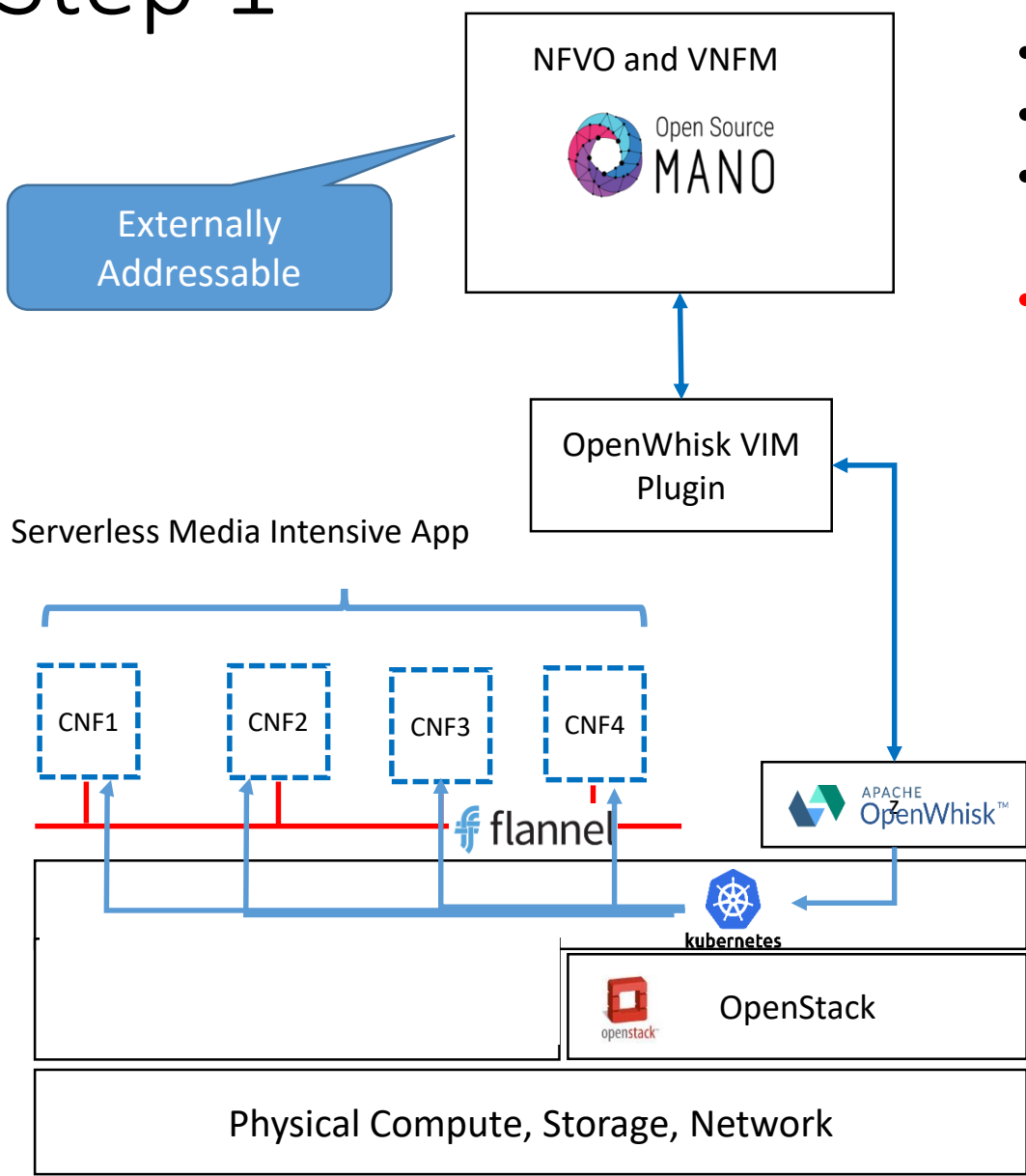


Figure 5-1: Multi-access Edge Computing framework

Source: Multi-access Edge Computing (MEC); Framework and Reference Architecture, ETSI GS MEC 003 v2.1.1 (2019-01)



Step 1



- Initially, we created OSM OpenWhisk Plugin
- This allowed to leverage K8s networking and scheduling
- While maintaining high level of FaaS abstraction (via OpenWhisk)
- **But then we understood that we cannot really orchestrate serverless applications with MANO**

What we needed to do was:

- To start some functions upon service instantiation (potentially nothing)
- Upon some event (e.g., in-app) start more functions
- Configure the rest of the service to work with them
- When they finish, they terminate
- Do it again and again with custom management flows
- To have K8s native experience

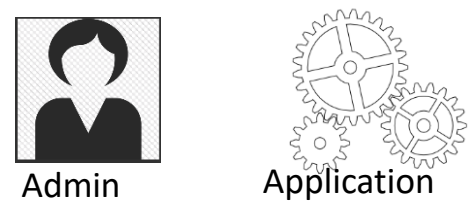
What we had with MANO was:

- No real modeling concept for serverless
- No custom management flows
- No natural hooks to cater for events
- No K8s devops experience
- **And increasingly it felt like an exercise in futility:**
 - **Trying to orchestrate containers from outside better than K8s**

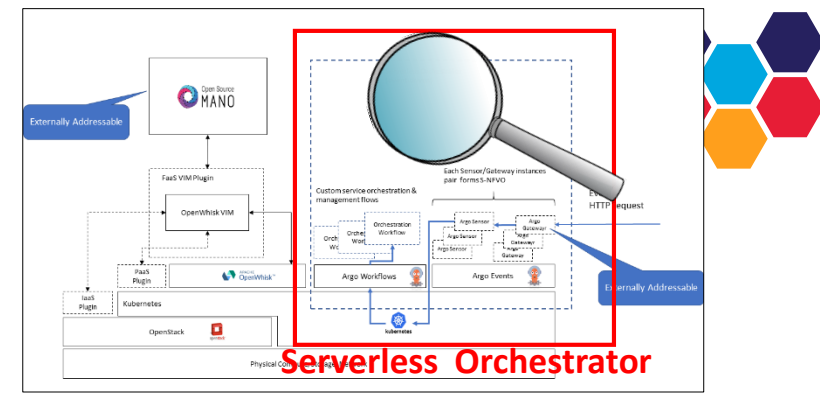


Argo Workflows & Argo Events to the rescue!





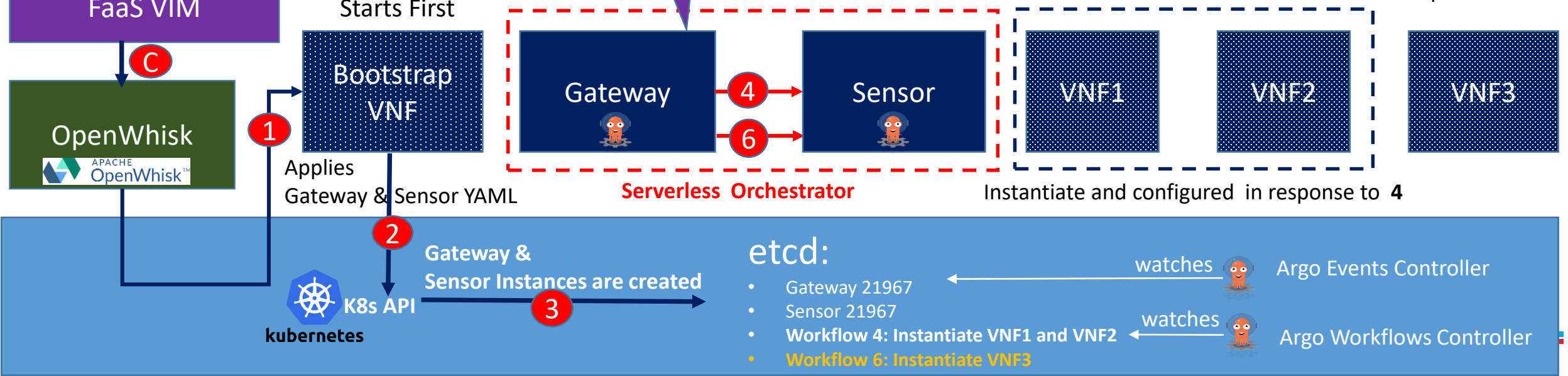
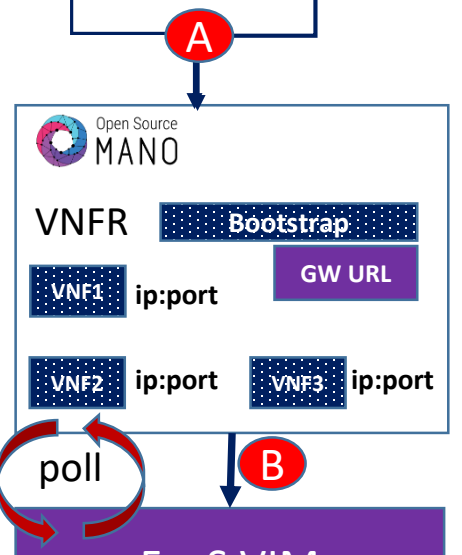
- Instantiates Onboarded Network Service comprising
- Bootstrap (special VNF to setup serverless Orchestrator)
 - VNF1 (FaaS)
 - VNF2 (FaaS)
 - VNF3 (FaaS)



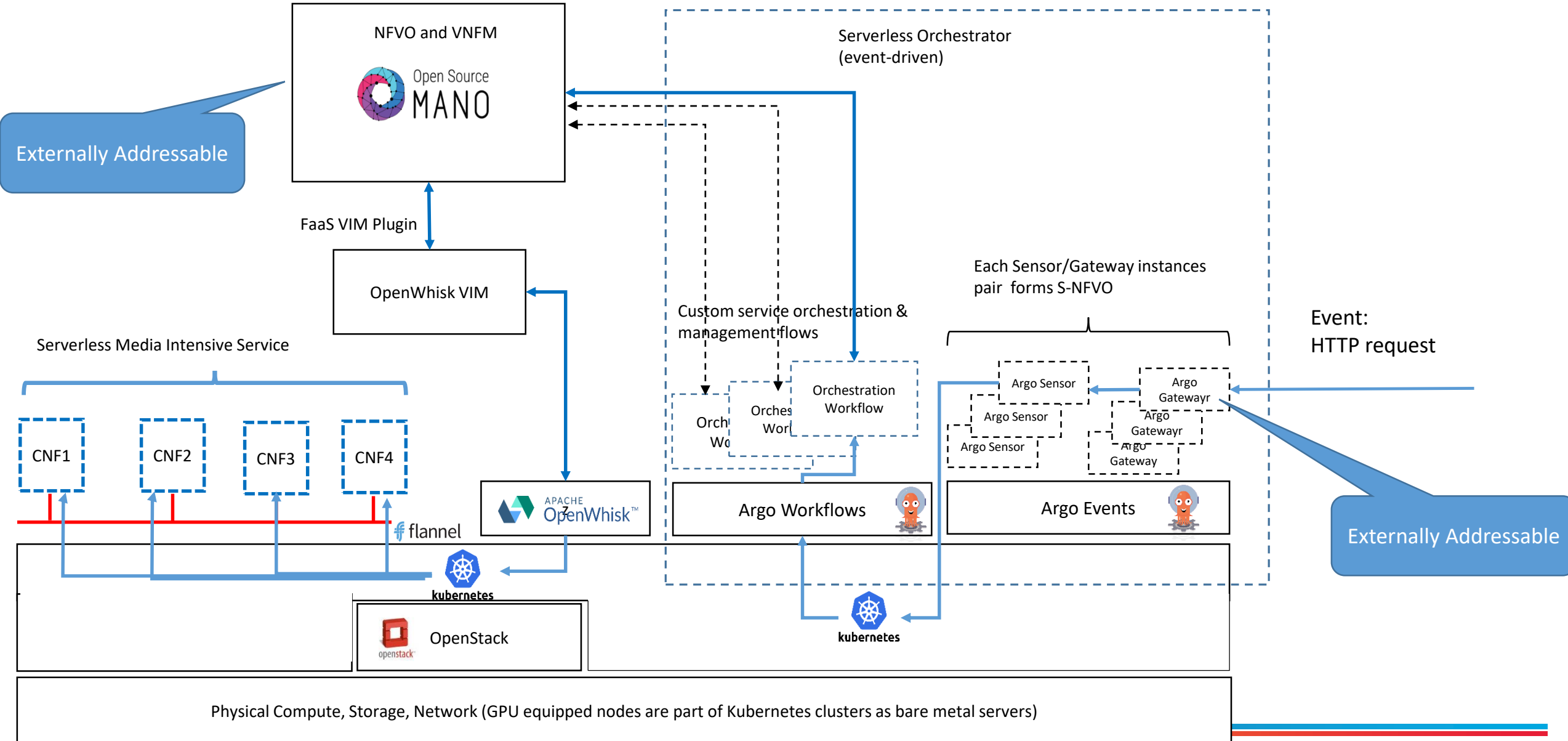
Request 1: instantiate VNF1 and VNF2

Request 2: instantiate VNF3

Serverless SO Endpoint (URL) for this service



Step 2:



Step 3: making GWs and Sensors serverless

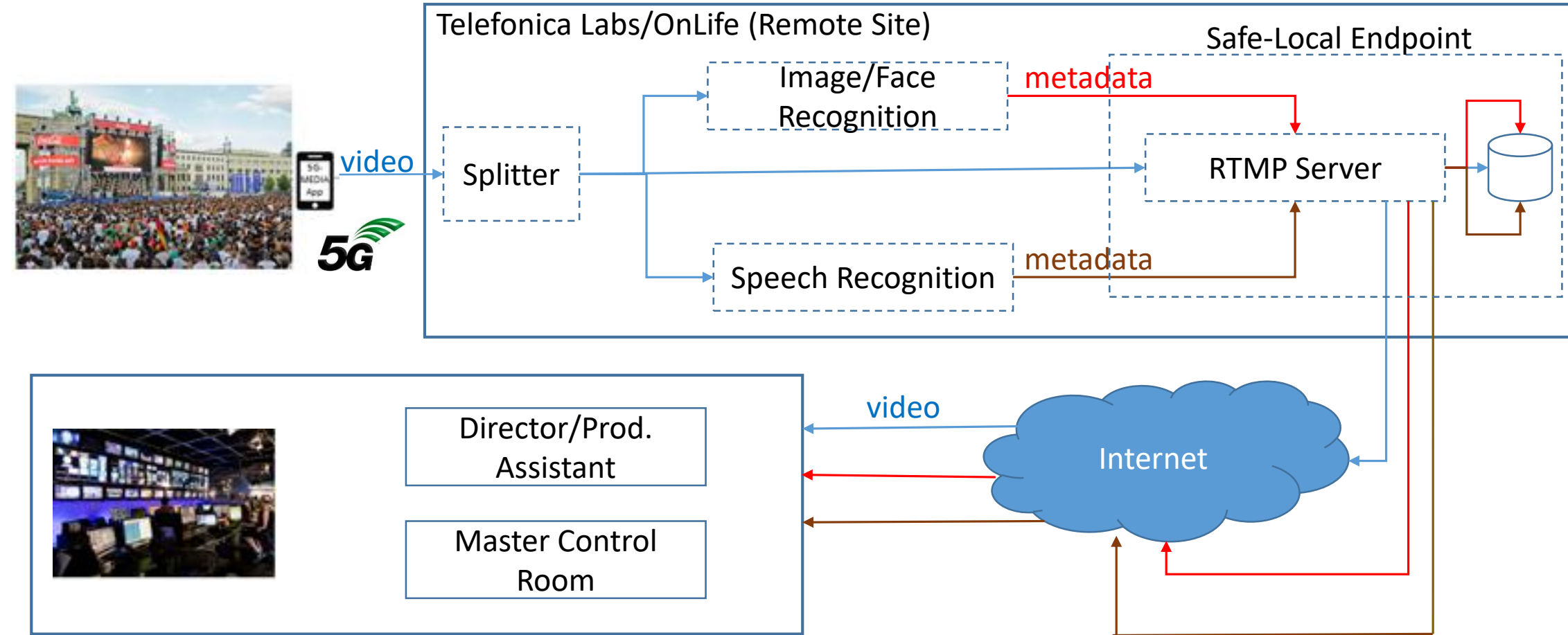


- GWs and Sensors proliferate in this design – Knative to the rescue!
- We forked Argo Events
- Made Gateway and Sensor Knative services
- So, if there are no events to orchestrate, they would not consume resources
- You can see it here: <https://github.com/IBM-Cloud/argo-events/tree/v0.11>





Just to recap...



- Collaboratively with 5G-MEDIA SVP, the application selects a most suitable edge for a contribution
- With serverless, production support functions are instantiated when and where needed, for the exact time they are needed
- Production support services are dynamically orchestrated

Conclusions



- Argo was extremely helpful for our project:
 - We found it easy to master
 - Very natural with nice declarative style
 - Really rich in features
 - Perhaps making Argo Events serverless can be an attractive feature?
- Do we really need any orchestration engine besides K8s?
- Maybe we should treat it as what it is – a very smart orchestrator rather than VIM?
- Workflows vs Operators:
 - Workflows relieve us from boilerplate code and Argo excels in this!
 - Operators try to reconcile in a true cloud-native style
- So, should we pick one or can they complement each other?



This work has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 761699.