**XGVela | What is XGVela?**

**Time**
- Launch: April 30, 2020
- First Meeting: May 21, 2020

**Status**
- Currently LF unfounded project

**Project definition**
An open source cloud native PaaS for applications and telco network functions, which is to enable new services and help mobile operators to seize the business opportunity from vertical industries in the 5G era.

**Name explanation**
- XG: 4G, 5G, 6G …… XG
- Vela: Sail in Latin to accelerate cloud native transformation and innovation

**Project links**
- Wiki: [https://github.com/XGVela/XGVela/wiki](https://github.com/XGVela/XGVela/wiki)
- Meeting: [https://github.com/XGVela/XGVela/wiki/XGVela-Meetings](https://github.com/XGVela/XGVela/wiki/XGVela-Meetings)
- Mailing list: [https://lists.xgvela.org/g/xgvela-tsc](https://lists.xgvela.org/g/xgvela-tsc)
- Partners: [https://github.com/XGVela/XGVela/blob/master/Partners%20interested%20in%20XGVela.pdf](https://github.com/XGVela/XGVela/blob/master/Partners%20interested%20in%20XGVela.pdf)
- Current TSC: ZTE, CMCC, Nokia, Intel, Ericsson, Huawei, Wind River
1. **Application tailoring:**
   - The NFs / applications are further decomposed according to the microservices architecture.
   - Strip away the parts that have nothing to do with the application itself.

2. **Platform addition:**
   - Support the coexistence of multiple resource forms.
   - Based on network element software architecture, the implementation of the general service rely on the platform.
   - Provides unified capabilities through API.

---

**XGVela | How to achieve XGVela**

**Telco-PaaS** (under exploring)

- HA
- Charging
- O&M
- IPSec
- DPI
- SIG
- ...  

**General-PaaS**

- service registry & discovery
- Unified Platform Architecture
- unified communication mechanism
- common database
- common middleware
- big data analysis/AI
- service LCM

**Heterogeneous IaaS**

- x86
- ARM
- Acceleration
- Storage
- Networking

---

**SMF**

- policy execution
- session management
- Telco common service logic
- software structure

**App**

- App specific logic 1
- App specific logic 2
- Telco common service logic
- software structure

**Light SMF**

- session management
- policy execution

**Light App**

- App specific logic 1
- App specific logic 2

---

1. Application tailoring:
   - The NFs / applications are further decomposed according to the microservices architecture.
   - Strip away the parts that have nothing to do with the application itself.

2. Platform addition:
   - Support the coexistence of multiple resource forms.
   - Based on network element software architecture, the implementation of the general service rely on the platform.
   - Provides unified capabilities through API.

---

**HA**

**Charging**

**O&M**

**IPSec**

**DPI**

**SIG**

---

**Telco common service logic**

**App specific logic 1**

**App specific logic 2**

**Software structure**

**Unified communication mechanism**

**Common database**

**Common middleware**

**Big data analysis/AI**

---

**Hypervisor**

**Computing/storage/network**

**Container**

---

**General-PaaS**

- service registry & discovery
- Unified Platform Architecture
- unified communication mechanism
- common database
- common middleware
- big data analysis/AI
- service LCM

---

**Telco-PaaS** (under exploring)

- HA
- Charging
- O&M
- IPSec
- DPI
- SIG
- ...

---

**Light SMF**

- session management
- policy execution

**Light App**

- App specific logic 1
- App specific logic 2

---

**SMF**

- policy execution
- session management
- Telco common service logic
- software structure

**App**

- App specific logic 1
- App specific logic 2
- Telco common service logic
- software structure

---

**SMF**

- policy execution
- session management
- Telco common service logic
- software structure

**App**

- App specific logic 1
- App specific logic 2
- Telco common service logic
- software structure

---

**Hypervisor**

**Computing/storage/network**

**Container**

---

**Heterogeneous IaaS**

- x86
- ARM
- Acceleration
- Storage
- Networking

---

**Hypervisor**

**Computing/storage/network**

**Container**

---

**Heterogeneous computing as a Service**

- x86
- ARM
- Acceleration
- Storage
- Networking
Focus on telco platform

- Define a set of functions (PaaS service for reuse across CNFs & applications) needed in telco platform
- Use cases & requirements
- Choose proper software to integrate telco platform
- Development & enhancement of functional components
- Interface: how to provide service to users and applications
- POC
XGVela | Relationship with other Communities/Organizations

- **Ref Standards**: 3GPP, NGMN
- **Network Functions & Applications**: A 5G Cloud Native PaaS
- **Orchestration**: XGVela
- **Integration & Testing**: Kubernetes, ONAP
- **PaaS & Platform**: Cloud Native Computing Foundation

Cooperate to achieve orchestration including PaaS
Cooperate to achieve best practice
XGVela | Relationship with CNCF

- XGVela integrates necessary CNCF projects to form General-PaaS.
- Telco enhancement requirements on General-PaaS will be explored and contribute to CNCF project.
XGVela | Relationship with CNTT

- **Difference between CNTT RA2 & XGVela**
  - ✓ CNTT RA2 scope: Kubernetes capabilities, support CNFs (pure container or VM & container mixed)
  - ✓ XGVela scope: PaaS capabilities, support telco NFs/applications development

- **Potential coordination between CNTT & XGVela:**
  - ✓ To support telco NFs
    - ✓ XGVela picks base platform based on CNTT RA2, including CNTT RI2, ICN (Akriano BP) etc
    - ✓ More implementation details will be discussed after XGVela telco PaaS definition and user story pick
  - ✓ For new requirements on CaaS in XGVela implementation
    - ✓ Implementation would either exist as plug-in/extension, or upstream back to original project
    - ✓ Requirements would firstly be contributed to CNTT RA2 (trying to maintaining a unified telco CaaS base)
XGVela | Relationship with ONAP (1/2)

• **Difference between ONAP & XGVela**
  ✓ ONAP scope: lifecycle support of NFs/services, standard NF/service management from different vendors, environment management
  ✓ XGVela scope: PaaS capabilities, support telco NFs/applications development

• **Potential coordination between ONAP & XGVela:**
  ✓ To support NFs/services:
    ✓ XGVela enables PaaS abilities & functional modular to be orchestrated by ONAP
    ✓ XGVela together with ONAP to design end-to-end management including PaaS
  ✓ To support ONAP microservice architecture:
    ✓ XGVela integrate cloud native toolsets help to implement ONAP microservices
    ✓ XGVela takes ONAP as an application and a requirement input
Community:
• Apr 30, launch as LF project
• May~July, get XGVela running: clarify project goals & establish working group
• Dec, Release 1
• Try to join LFN after project is stable

Testbed:
• Build a prototype of cloud native telco-platform

Demo/POC:
• A demo involving XGVela & network functions (details story to be discussed)

Telco PaaS definition at:
• https://github.com/XGVela/XGVela/blob/master/XGVela%20functions%20%26%20implementations.xlsx

Next: use case discussion
Thank you
With the help of NFV, SDN and orchestration management technology, current operator network is transforming from the traditional hardware and software equipment to the layered and decoupled cloud network.

In the future, thanks to the application of container, microservice and other technologies, it will eventually evolve into the cloud native network.

### Operators’ Network Transformation

**Core Network**

**Access Network**

**Business Platform**

**Decoupling**

**Cloudified network**

**Cloud native**

---

**Virtualization Layer**

- 4G
- 5G
- brs

- Compute
- Storage
- Network

---

**Cloud native**

- NF
- NF
- NF

**Virtualization/Container Layer**

- LB
- DB
- Billing

---

**PaaS**

- Compute
- Storage
- Networking
XGVela | Why need cloud native telco PaaS (2/2)

Fast-paced change in 5G requirements

- High flexibility in 2B scenarios
- Quick function upgrade
- Agile capabilities release

VM platform is inadequate

- Guest OS cumbersome
- Low deployment density
- Slow start and stop of virtual machine

Open & healthy eco-system

- Reduce barriers to enter the telco-industry
- Expand and prosper the ecosystem
- Reduce the cost on network construction

Simplified design & innovation

- Carefully selected common services from NE to platform
- Standard APIs to provide capabilities
- Application development focus on service logic