AKRAINO

WHITE PAPER:

Akraino APIs

Table of Contents

Introduction

Edge computing is distributed computing with different application components interconnected by a network. Most applications we use are distributed: a smartphone with a cloud backend, a Nest Thermostat connected to the cloud, an Apple Watch connected to an iPhone and then to the cloud. We think of the "edges" as where these application bits run.

The obvious, existing edges are the "smart device" edge (e.g., the smartphone, the thermostat) and the public cloud. The simple value of the Internet edge is that we can do lots of things in the cloud that don't make sense on a device (the cloud resources are practically unlimited; the device resources vary limited as long as the connection is adequate; the collective behavior can be configured, tracked and managed from all devices). The limitations of the public cloud are the location of the data centers and the fact that the resources are shared.
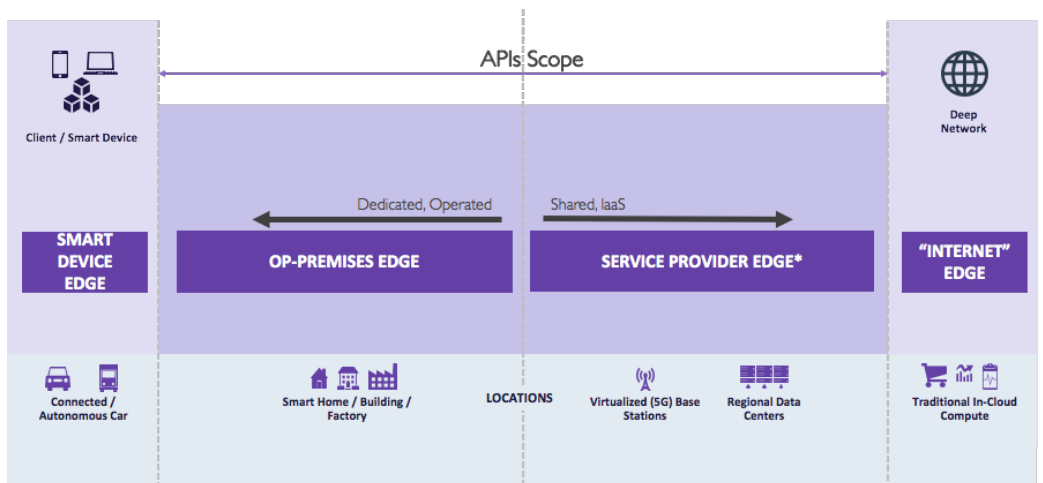
The newest is the telecom/5G edge — services delivered within the global telecom/ mobile infrastructure. It is usually a service, like the public cloud (although there are a growing number of private cellular systems with "premises" like values), it's more secure and private than the public cloud (because of differences between the Internet and cellular systems). And it leverages the massive trillion-dollar existing investment which already has hundreds of thousands of "edge" computers in place.
The challenges across Service Provider edge projects relate to distributed workloads processing and possibly mobility. Please refer to Linux Foundation Architecture white paper for more details on Service Provider edge.

What we need is an end to end service creation framework where the developers can write new edge applications: a software development and deployment platform which offers an open API Library for any developer to discover and consume the services the edge has to offer. We can learn from cloud models today how to develop, orchestrate and manage workloads/applications and offer developers similar continuous delivery (CD) pipeline functionality. Not to say that as devices and machines come online all the needs will be met, but at least it provides a framework to look at.

AKRAINO

In principle whoever is developing a solution for edge will appreciate resources scaling desired to address power limitation near the far edge Service Provider Radios. Hence the dynamic workload, specialized resources for acceleration, server less models are emerging in most of the Akraino Blueprints. Over last year we learned a few use cases, but we would like to elevate one of them amongst the pandemic unrest we are facing. China showed the world containment strategy by running workloads on 5G edge: predicting infected patients using a combination of facial recognition and advanced ML at edge.

In the near future we would like to present one of them, facial recognition utilizing open sourced TensorFlow, as use case for 5G/Edge along with the Architecture and APIs for continued learning in the community to develop open APIs for the next generation of developers to write applications for the next generation of internet, the 5G edge.

This paper is the first attempt to make available a generic set of edge APIs available for developers used to cloud native applications. As new infrastructure and Akraino blueprints evolve, there will be further revisions to these APIs.

Developer Edge APIs

This API set details a comprehensive Service Provider edge discovery and mobility requirements for commercial applications to operate across distributed sites. Device side APIs allow the control exchange to the Application Discovery Engine, from the client device, to register and find the nearest best compute instances to deliver the content.
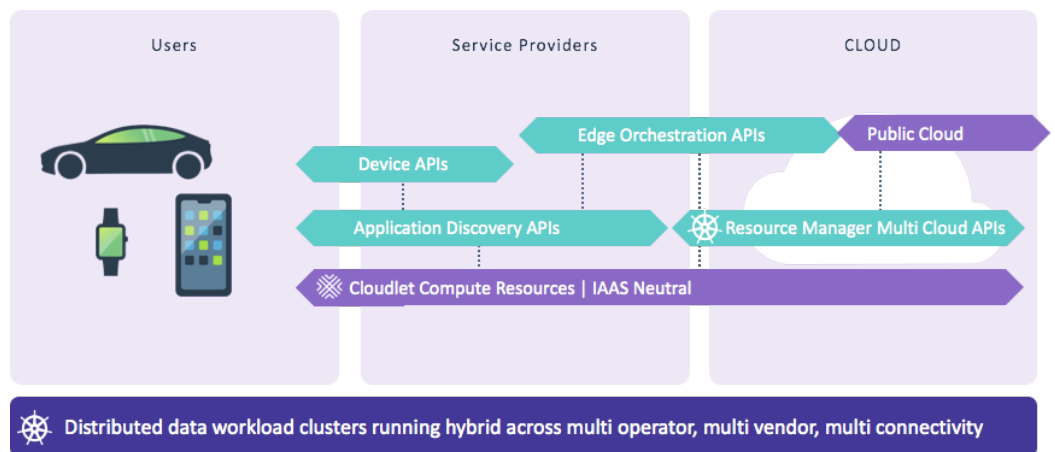
Thus, we have bucketed API sets shown above under edge orchestration APIs and Application Discovery APIs which coordinate with Client-side APIs.

LF EDGE

Edge Orchestration APIs

The Edge Orchestration APIs enable developers to define privacy, acceleration, and the Quality of Experience demands their applications may encounter declaratively with the cross cloud-native structure they require. For the Service Provider edge we structured two views.
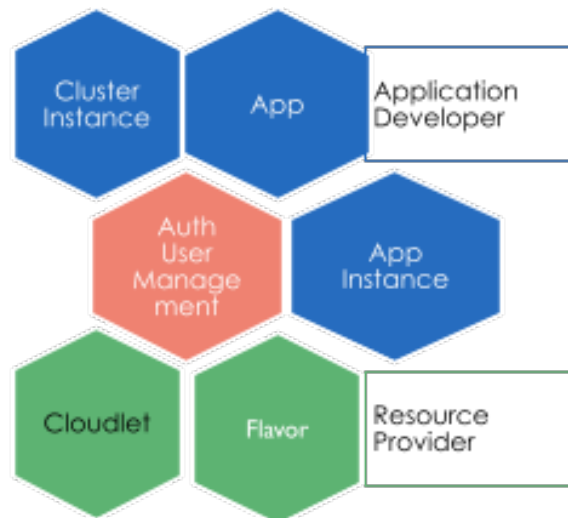
We have bucketed the edge orchestration APIs under two categories, Application Developer and Resource provider. As application developers need to evolve around not only deploying their workload on edge, but also needs for security and Role- based Access Controls. The resource provider needs related to auto onboarding of Infrastructure and managing their edge DCs, called cloudlets, and defining size of their offering via Flavors for developers to choose.

For Operator-related API, please visit the following link: https://api.akraino.org/#tag/Cloudlet

Developer and Application Onboarding Workflow

The onboarding workflow above defines orchestration of edge applications and provides several services to both application developers and operators. For application developers, these APIs allow the management and monitoring of deployments for edge applications. For infrastructure operators, these APIs provide ways to manage and monitor the usage of cloudlet infrastructures. Both developers and operators can take advantage of these APIS to manage users within the organization.

⊓⊔LF EDGE
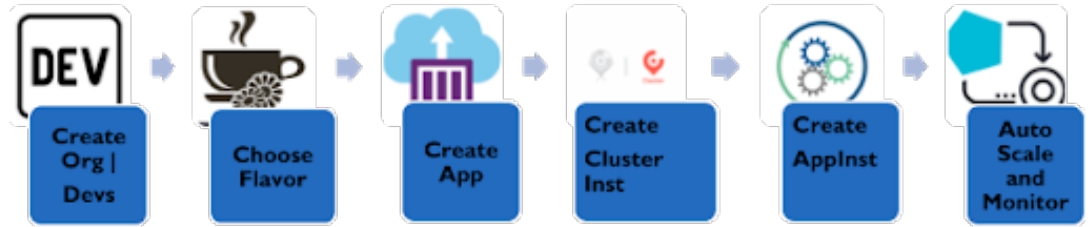
If you prefer to manage these services programmatically, the available APIs and their resources are accessible from the right navigational menu at links mentioned below:

For Auth and User Management Please follow these links:

Authentication: https://api.akraino.org/#tag/Security
Role Based Access Control:
Organization: https://api.akraino.org/#tag/Organization
Developer: https://api.akraino.org/#tag/User

For more details on APIs definitions related to app lifecycle management, please follow the links below for edge app deployment:

Create Cluster - Microservices deployed as a set of containers or VMs:
https://api.akraino.org/#tag/Cluster-Instance

Create App Manifest - Define app mobility strategy: QoE, Geo store | Privacy policies:
https://api.akraino.org/#tag/Application

Create App Instance: Launch application backend and auto scale:
https://api.akraino.org/#tag/Application-Instance

Application Discovery Engine (ADE) APIs

The primary function of the ADE is to match an end-user with an application instance. The matching process is initiated once the end user's device registers via SDK (https://api.akraino.org/device/#operation/RegisterClient). The ADE receives this API call, searches an inventory of application instances, and enables the end-user device to establish an encrypted communication channel with the closest instance. If the user is connecting via a 4G LTE or 5G network, then the ADE is capable of validating the legitimacy of the subscriber and their network access permissions through integration with the network operator's operations and billing infrastructure.

As the promise of edge is ultra-low latency, the Service Provider edge will have to provide a deterministic way to measure and enforce quality of experience based on application needs primarily being latency, bandwidth. Assuming the above promise it entails embedding ADEs across operators which not only records the application backends

🔲LF EDGE

nearness, but also health and QoE/App across all edge sites within a region, and offers a control API to fetch the best location and tune the application back-end to deliver the best experience.

The Control API from the discovery engine returns the ranked list of Uniform Resource Identifiers (URIs) identifying application back-end within sites nearby.

URI selection Criteria:

• App instances in sites Geolocated based on Clients location
• URI rank based on recent L4 QoE (Latency, Bitrate) measurements
For more information on definition of this control API, please visit:
https://api.akraino.org/device/#operation/FindCloudlet

Location verification relies on calls to APIs provided by telecom operators, and the availability and accuracy of location data will vary from operator to operator. Additional parameters, such as application subscription and service contracts among the mobile network operators, can be taken into account during the matching process.
https://api.akraino.org/device/#operation/VerifyLocation

Client-Side APIs

https://github.com/mobiledgex/edge-cloud-sampleapps

The link above provides sample apps along with a set of SDKs and Libraries containing a collection of operations and resources that assist in registering and discovering the best application instance running on a Service Provider edge cloud.

Application Discovery Engine SDKs and Libraries are available in different languages and platforms, which include Android, Unity, and iOS, allowing for easy application development.

The Performance Metrics API, provided with each SDKs, tracks the average latency characteristics of the edge network. To take advantage of this API, you must add your application server's Application Instance to a list of sites in order to periodically monitor them.

□LF EDGE

Blueprints and Integrations with other LF Edge projects

Figure below shows how these APIs are related to Akraino blueprints.
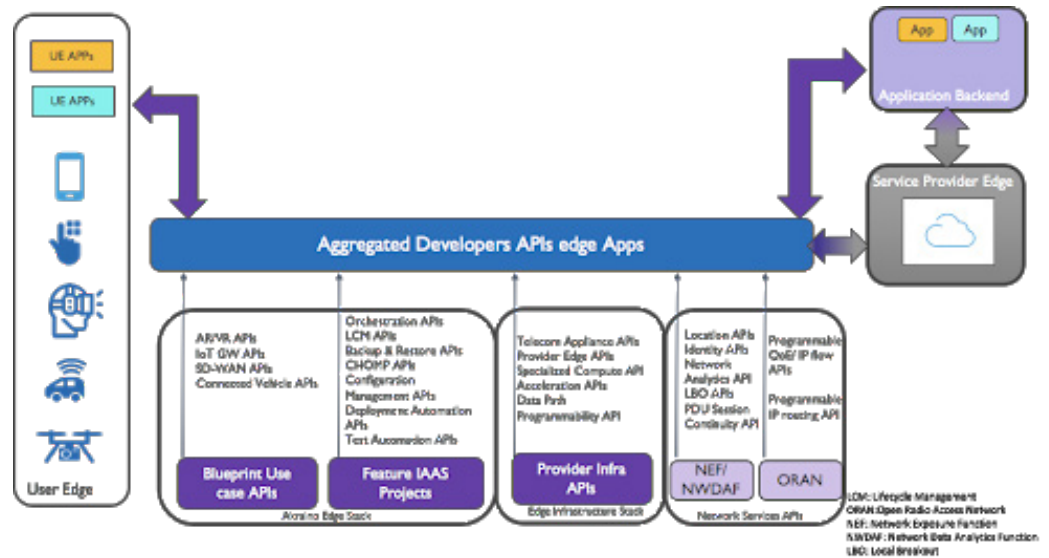
Akraino R1/R2 blueprints offer a rich set of APIs for a variety of Edge use cases such as AR/VR, Connected Car, IoT Gateways, SD-WAN to name a few. These APIs include APIs for Edge deployment automation, configuration management, life cycle management as well as APIs to support SD-WAN, AR/VR, Connected Car, Provider Edge and Telecom Appliance use cases. Besides, Akraino Feature projects such as Regional Controller, Backup & Restore, CHOMP and Blueprint Validation provide support for Orchestration APIs, backup and restore APIs, Cluster Health and Overload Monitoring APIs, and Test Automation APIs.

The API framework shown in the figure above can integrate Infrastructure APIs as well as APIs from network services like SCEF and NEF from the 3GPP Mobile Core. Further, as public cloud providers deploy their edge instances, Akraino APIs can provide many value-added services. The need to interface and exchange information through these APIs will allow competitive offerings for consumers, enterprises and vertical industry user segments.

The API Aggregation and Management layer will be implemented as a Feature project in R3/R4 releases. This will ensure all Akraino blueprints will comply with these APIs.

Glossary

Abbreviations	Description
API	Application Programming Interface
5G	5th Generation Mobile Network
IoT	Internet of Things
GPU	Graphic Processing Unit
QoE	Quality of Experience
UE	User Equipment
AR	Augmented Reality
VR	Virtual Reality
DC	Data Center
IAAS	Infrastructure as a service
PAAS	Platform as a service
SAAS	Software as a service
Application Backend	Software written by a developer that interacts with an end user's device, such as a mobile smartphone, augmented reality glasses, autonomous vehicle, or other such devices.
Application Instance	A single deployment of an application backend
Cloudlet	A point of presence for application deployment that is composed of compute, networking and storage infrastructure. Cloudlets within public telecom operators' networks are often provisioned on top of servers and a virtual infrastructure layer managed by the operator; and interface with the operator's business and operations support systems.

LF EDGE

Abbreviations	Description
Cluster Instance	A Kubernetes cluster with one or more container run-time instances operating inside of it. To provide robust isolation between users, Kubernetes clusters are provisioned within separate virtual machines or hosts.
Flavor	A compute instance with a specific amount of virtual CPU cores, system memory (RAM), storage and optionally, an attached GPU. Flavors can vary between operator networks.
Operator	A supplier of one of more cloudlets
Developer	A user of the APIs who can (i) upload and manage application images, (ii) create clusters and (iii) deploy applications and instances. Developers are members of organizations and can have varied administrative capabilities. Only the developers who are members of a particular organization can see the applications and other objects within it.
Organization	A group of users associated with applications and application deployment policies. When an organization is created it automatically provisions image registries for exclusive use by the organization
ADE	Application Discovery Engine, The primary function of the ADE is to match an end-user with an application instance. The matching process is initiated once the end user's device sends a registerClient API call via client SDK. The ADE receives this API call, searches an inventory of application instances, and enables the end-user device to establish an encrypted communication channel with the closest instance.

Upon calling the supplemental findCloudlet API, the end user's application is matched with the closest application instance satisfying the desired QoE.

For the findCloudlet API call to succeed, the application backend is instantiated on at least one of the cloudlets within the operator network in which the end user's device is connecting. It is the responsibility of the developer to choose the initial cloudlet (location) in which the application backend is deployed. |