Home Edge Computing?

In home appliances, the available H/W resources (CPU, Memory, Storage, etc.) are limited, but the user services require more and more resources. Single device is difficult to provide high-quality services due to computing resource constraints.
**Tech. Requirement**

The Home Edge Project will offer an edge computing open source platform for various use cases in the home, meeting the technical demands of products and services such as:

› Dynamic device / service discovery at Home Edge
› Distributed machine learning
› Multi-vendor interoperability
› Security and data privacy of users
› QoS guarantee in various dynamic conditions (e.g. devices On or Off)
### Requirements: Home Edge

<table>
<thead>
<tr>
<th>Key Features</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Edge Orchestration** | - Find edge devices/services  
- Monitor device resource to choose best one  
- Service offloading  
- Service lifecycle management |
| **Data Storage**     | - Collect and share data for home devices  
- Provide persistent storage |
| **Home Device Control** | - Discovery home devices in local network  
- Control home devices  
- Provide interface between cloud and home devices |
| **Machine Learning** | - Execute neural network model using neural network Framework  
- Allocate distributed job and manage scheduling |
| **Security**         | - Provide function to secure peer to peer communication  
- Protect user data |
High-Level Platform Architecture

LF Home Edge Apps and Services based on APIs

- **Edge Setup**
- **Data Service**
- **Device Control Service**
- **Other Edge Services...**
- **Speech Recognition**
- **Vision Service**
- **Other AI Service**

**Legend**

- **Edge Service**
- **Device Control**
- **AI**
- **Runtime Engine**

**Security**
- Edge Orchestration
  - Configuration Manager
- Discovery Manager
- Scoring Manager
- Edge App Manager
- Secure Manager

**Data Storage**
- I/O Agent
- Command
- Metadata
- Core Data

**Home Device Control**
- Cloud Interface
- Device Setup
- Controller Adapter
- Air Conditioner

**Machine Learning**
- NN Model Interface
- Model/Data Partition Converter
- Distributed Job Scheduler
- Job Executer

**Container Runtime / Deep Neural Net Runtime**

* Samsung seed code is based on EdgeX Foundry that is able to provide real-time, locality, and user privacy for various use cases, initially focused on Orchestration and storage.
Module Description (1/5)

**Edge Orchestration Module**

› **Configureration Manager:**  
Services need to register their information to use edge orchestration. These info is used to process each distributed service.

› **Discovery Manager:**  
Find the home edge devices and their services. Default is local network. NAT Traversal also can be supported

› **Scoring Manager:**  
Check the available (computing) resource and scoring for each registered service
Module Description (1/5)

Edge Orchestration Module (Cont’d)

› **Edge App Manager:**
Handle command to execute application on a remote device and notify status. Manage executed application’s life-cycle.

› **Secure Manager:**
Encrypt & decrypt data between orchestration to orchestration. Provide simple method to identify user’s device.
Module Description (2/5)

Data Storage Module

› **Core Data**:
  Provides a persistent storage on the Home Edge device for those collected data from its connected home devices (CE device, sensor, thing) and services.

› **Metadata**:
  Has knowledge about the ID / profile / data of the connected home devices (CE, sensor, thing) and services.

› **I/O Agent**: Provides an API for accessing the data storage.

› **Command**: Provides a method for getting specific data from the data storage of the Home Edge device by rule-based approach.
Module Description (3/5)

Home Device Control Module

› **Cloud Interface**: Provides the interface between cloud services and each home device in a user's home network.

› **Device Discovery**: Finds the connected home devices from the Home Edge device, when the connected home device is found for the first time, their client service will be installed by Controller Installer.

› **Device Setup**: Sets the required network information of the connected home device for making it join a user's home network.
Module Description (3/5)

Home Device Control Module (Cont’d)

› **Controller Installer**: Installs a client service of the discovered connected home device.

› **Controller Adapter**: Provides unified APIs for the control of the connected home devices.

› **Home Device Client**: Controls the connected home devices (e.g. user's refrigerator, washing machine, light bulb, door-lock, HVAC, and temperature sensor).
Module Description (4/5)

Machine Learning Module

› Neural Network Model Interface: Provides APIs for inference and recognition using (distributed) neural network processing.

› Model Partition Converter: Divides model for distributed neural network processing.

› Distributed Job Scheduler: Schedules and allocates distributed job.

› Job Executer: Executes distributed job on the Home Edge device using Deep Neural Network Framework.
Module Description (5/5)

Security Module

› Security Module provides security features that the Home Edge devices and services should have to provide, such as secure on-boarding, Certificate, AAA, encryption/decryption of the message protocols, and so on.

Deep Neural Network Framework

› Deep Neural Network Framework provides dataflow programming framework for machine learning such as TensorFlow Lite.
High-Level Platform Architecture

Edge Setup | Data Service | Device Control Service | Other Edge Services... | Speech Recognition | Vision Service | Other AI Service

Container Runtime / Deep Neural Net Runtime

Legend

- Edge Service
- Device Control
- AI
- Runtime Engine

LF Home Edge Apps and Services based on APIs

- Edge Orchestration
  - Configuration Manager
  - Discovery Manager
  - Scoring Manager
  - Edge App Manager
  - Secure Manager

- Data Storage
  - I/O Agent
  - Command
  - Metadata
  - Core Data

- Home Device Control
  - Cloud Interface
  - Device Setup
  - Controller Adapter
  - Air Conditioner

- Machine Learning
  - NN Model Interface
  - Model/Data Partition Converter
  - Distributed Job Scheduler
  - Job Executer

- Deep Neural Network Framework (e.g. TensorFlow Lite)
Current Status & Coverage of Alpha Release

› Current Status
  › Configuration Manager: set configuration for registration
  › Discovery Manager: local network (using mDNS-SD)
  › Scoring Manager: measuring resources (CPU, Memory, NW Bandwidth)
  › Edge App Manager: remote service execution, notify status
  › Secure Manager: data encryption & decryption

› Alpha Release (~June. 2019)
  › orchestration modules are under verification
  › add more functionalities
  › handle exception cases
**Edge Orchestration Architecture**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge (Orchestration) Agent</td>
<td>Core service for performing functions of Edge Orchestration</td>
</tr>
<tr>
<td>Edge Device</td>
<td>Devices that provide services of Edge Orchestration</td>
</tr>
<tr>
<td>Edge App</td>
<td>Application using Edge Orchestration Agent</td>
</tr>
<tr>
<td>Device Resource</td>
<td>Resource available on the device (e.g. CPU, Memory, Storage, NW Throughput, NW Bandwidth)</td>
</tr>
<tr>
<td>Device Resource Scoring</td>
<td>Criteria for measuring of device idle resources (by each Edge App)</td>
</tr>
</tbody>
</table>

**Diagram Notes:**
- Edge Orchestration App
- Edge Orchestration Agent
- Orchestration Internal API
- Configuration Manager
- Discovery Manager
- Scoring Manager
- Edge App Manager
- Secure Manager
- REST Client (E2E)
- REST Server (E2E)
- IPC / Network
- Edge Orchestration Core (Go)
## Orchestration to Orchestration API

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discovery</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td>GET</td>
<td>/api/v1/discoverymgr/devices</td>
</tr>
<tr>
<td></td>
<td>POST</td>
<td>/api/v1/discoverymgr/devices</td>
</tr>
<tr>
<td></td>
<td>POST</td>
<td>/api/v1/discoverymgr/devices/TXT</td>
</tr>
<tr>
<td><strong>Edge Service (App)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>POST</td>
<td>/api/v1/servicemgr/services</td>
</tr>
<tr>
<td>Status</td>
<td>POST</td>
<td>/api/v1/servicemgr/services/notification/{serviceid}</td>
</tr>
<tr>
<td><strong>Scoring</strong></td>
<td>scorevalue</td>
<td>GET</td>
</tr>
</tbody>
</table>
RoadMap 2019 (Edge Orchestration)

<table>
<thead>
<tr>
<th></th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
<th>Jul</th>
<th>Aug</th>
<th>Sep</th>
<th>Oct</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LF Home Edge</strong></td>
<td>[Release] LF Home Edge Alpha</td>
<td>[Release] LF Home Edge v1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TODO</strong></td>
<td>1. Edge Device/App Discovery</td>
<td>4. Enable NAT Traversal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
RoadMap

<table>
<thead>
<tr>
<th>LF Home Edge</th>
<th>2019</th>
<th>2020</th>
<th>2021</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Orchestration - Docker Container - Local NW</td>
<td>Alpha (2019.06)</td>
<td>v1.0 (2019.10)</td>
<td>v2.0 (2020.10)</td>
</tr>
<tr>
<td>Edge Orchestration - Android - NAT Traversal - Sample Service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine Learning - Data Parallelism - Service Parallelism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Home Device Control - OCF Device</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interoperability with Cloud - Service Download</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edge Service (TBD) - Domain Specific</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LF Home Edge

F2F meeting during ONS
Edge Orchestration (1/2)

Assignment of “Master” edge device among multiple edge devices at home

› Sharing resource with all edge devices, when a new one is added to the network.
› Assignment of “Master” edge after analyzing current resource status and capability about all the edge devices.

<table>
<thead>
<tr>
<th></th>
<th>TV</th>
<th>Fridge</th>
<th>AI Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always ON</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>1.7 GHz</td>
<td>1.3 GHz</td>
<td><strong>2.2 GHz</strong></td>
</tr>
<tr>
<td><strong>Core Number</strong></td>
<td>4</td>
<td>4</td>
<td><strong>8</strong></td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>2.0 GB</td>
<td>4.0 GB</td>
<td><strong>1.8 GB</strong></td>
</tr>
<tr>
<td><strong>Idle Memory</strong></td>
<td>400 MB</td>
<td>700 MB</td>
<td><strong>1 GB</strong></td>
</tr>
</tbody>
</table>

< Example on Master Edge Device Assignment >

< Master Edge Device Assignment Process >
Edge Orchestration (2/2)

Service deployment/execution/scheduling based on real-time resource status

› Real-time resource monitoring of all edge devices at home
› Edge service searching and download stored in the cloud
› Service deployment/execution on the edge
Data Parallelism

Optimization on elapsed time of inference through multi edge devices at home

- Execution of the same ML service to multi edge devices
- Distributed transmission of input data into multi edge devices
- Data distribution transmission scheduling based on edge device resource

< Data Parallelism (Data Parallelism, using same model) >
Service Parallelism

Parallel service processing through multi edge devices
› Simultaneous execution of variant ML services into multi edge devices
› Transmission of all input data into multi edge devices
› Model deployment scheduling based on edge device resource

< Service Parallelism (using same data, different model) >
Model Parallelism

ML inference / model learning through multi edge devices
› Distribution and execution of a single ML model from multi edge devices
› Model partition w.r.t. the size of ML model, cost of communication, resource

< 4:2 Layer Partition >

< 3:2 Feature Partition Model >
Executive Summary

Where: Intel Santa Clara Office, 08:00 ~ 12:00 on April 2\textsuperscript{nd}.
Material: https://wiki.lfedge.org/pages/viewpage.action?pageId=4624102

Agenda

› Overview of Home Edge Project
› Use cases driving the design / implementation of Home Edge Project
› Deeper dive on the project connecting use cases with functionality
› Home Edge possible candidates or Blueprint
› Possible collaboration with other projects in LF Edge and members
› AoB
Highlight – Requirement Feedback (1/4)

**Edge Orchestration (Master Edge)**
- *Intelligent Master Edge Assignment*: Beyond policy → pattern thru learning.
- *Intelligent connectivity management*: to bridge the various connectivity at home.
- *Flexible negotiation*: for the Master Edge Assignment via given various situations.
- *Resource sharing including wireless (e.g. 5G) channel condition*: for optimized routing.

**Distributed & Parallel ML**
- *Learning from the Home Edge Device*: as well as from the cloud.
- *ML application expansion*: to devices control & contents subscription as a user’s pattern.
Highlight – Requirement Feedback (2/4)

Home Edge Profile
› A clearer guideline for minimum memory footprint & storage size.
› Definition & functionalities for Edge Device other than CE devices (e.g. doorlock).
› Multi-tier (level) Edge profile: and features, use cases, architecture, and common layer.

APIs
› API document (before seed code release): for the collaboration items in the community.
› Clarifying the API development and management relationship with EdgeX and others.

Data Privacy Policy
› Multi-level policy management: e.g. TV (contents – more sensitive) vs. A/C (temp. control).
Highlight – Requirement Feedback (3/4)

**Enabler with Telco Business (if necessary)**
- Possibility to propose a new Blueprint to Akraino: 5G Home Service use cases [WIP].
- Numeric verification to prove the tech. benefit: e.g. latency, computing power, etc.

**Container Runtime Alternatives**
- Support container runtime alternatives (or updates): possibility to co-work with EVE.
- OS and runtime definition for the better technical understanding.

**Ecosystem (Marketplace)**
- Container deployment infra. beyond DockerHub as marketplace (w/ Certification).
- Open platform: to grant 3rd party to access the Home Edge for enabling their services.
Operational

› **WG per use case** as well as the platform modules (technical): use case mining.

› Direction: collection of gap analysis, tech. requirement ➔ roadmap v2.0 ➔ v1.0 release
  
  › Designated pages for the gap analysis, tech. requirement gathering.