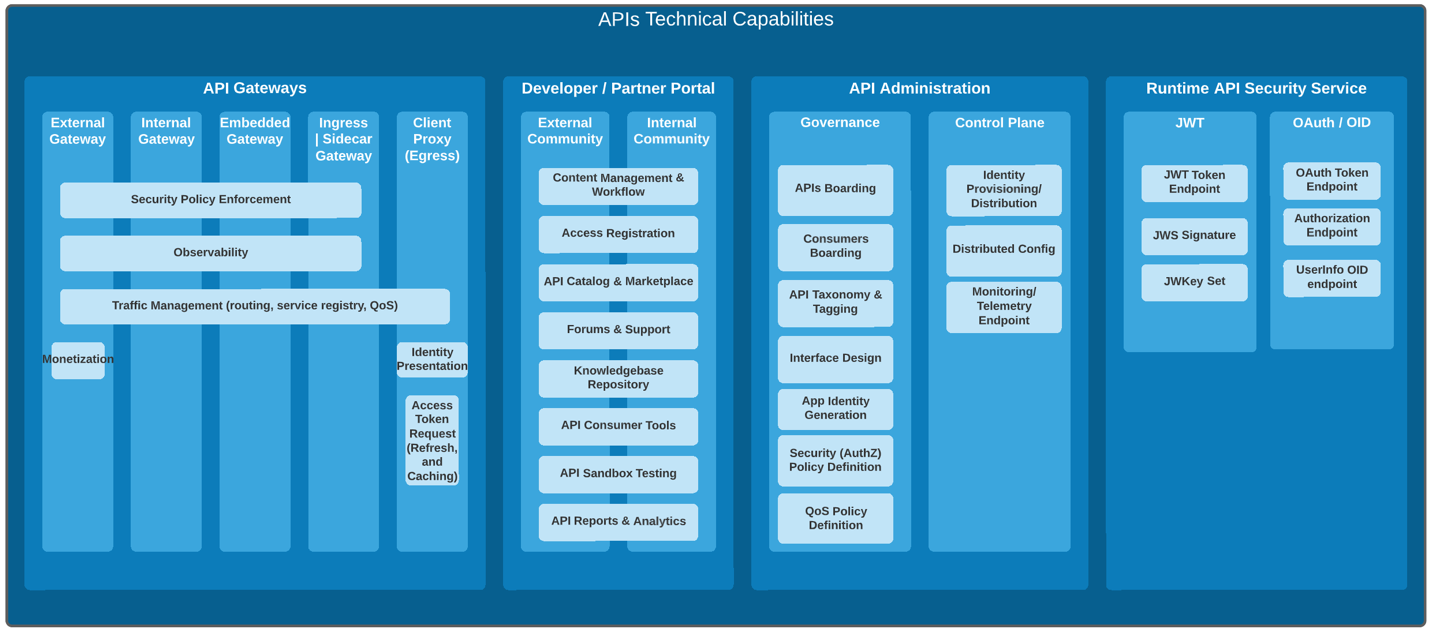
**API Reference Architecture Model**

API Technical Capabilities Model and high level Reference Architecture Model

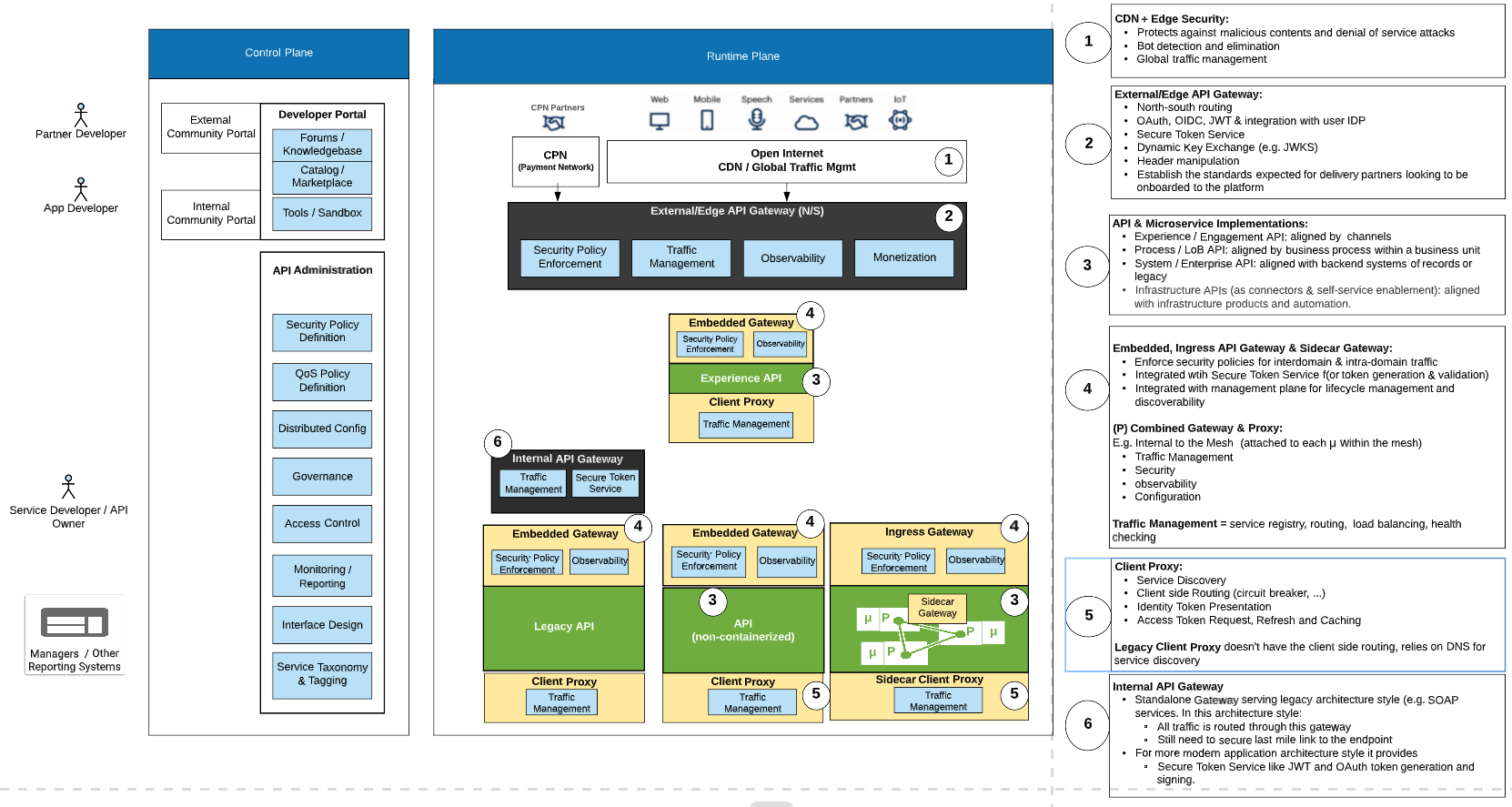
**API Technical Capabilities Model**

Below is 4-levels capabilities model outlined for the API application domain:



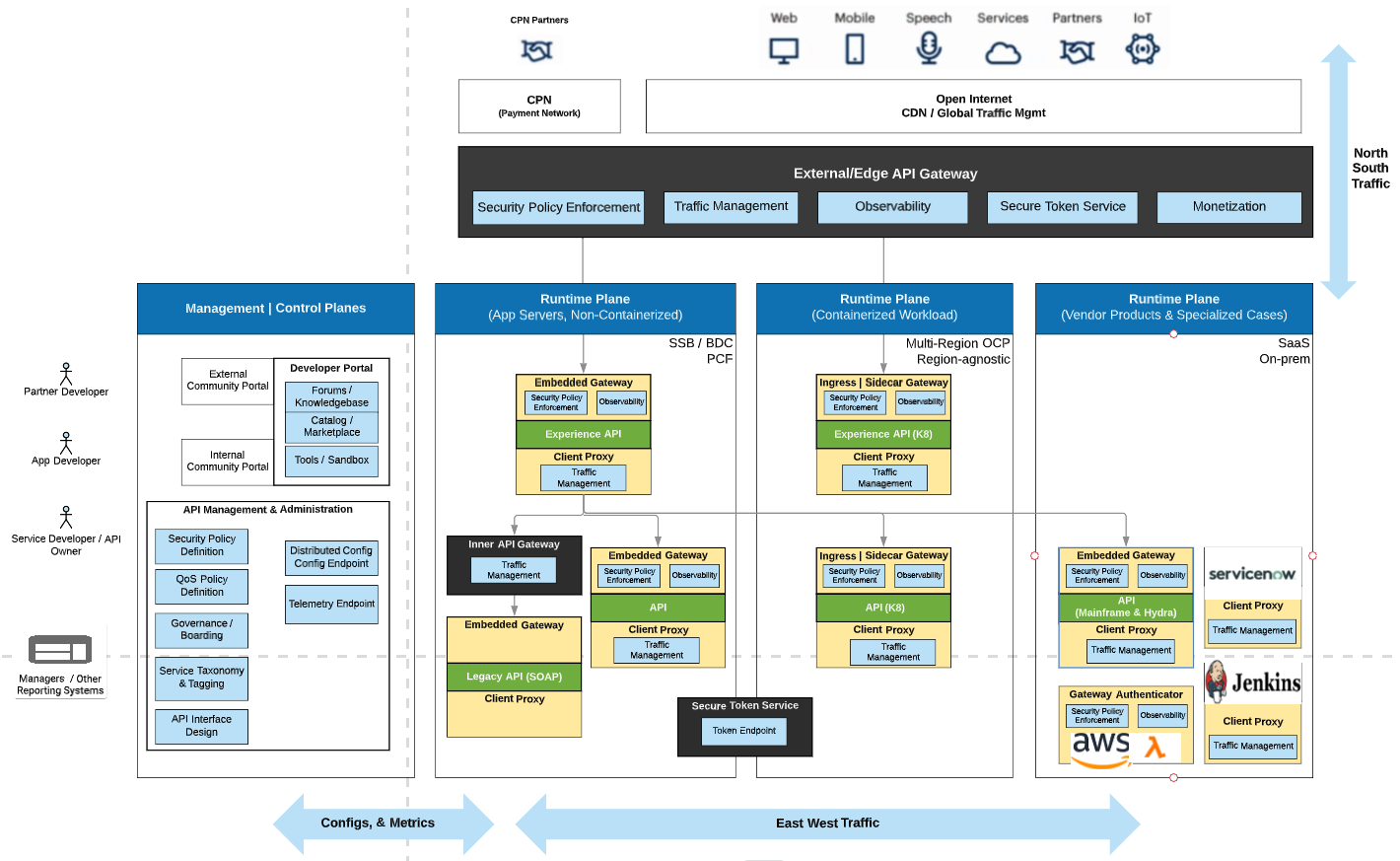
**API High Level Reference Architecture Model**

This is a high level reference architecture model showing where the various technical capabilites are provided and by which architecture component:



**API High Level Reference Architecture Model - Multiple Runtimes Viewpoint**

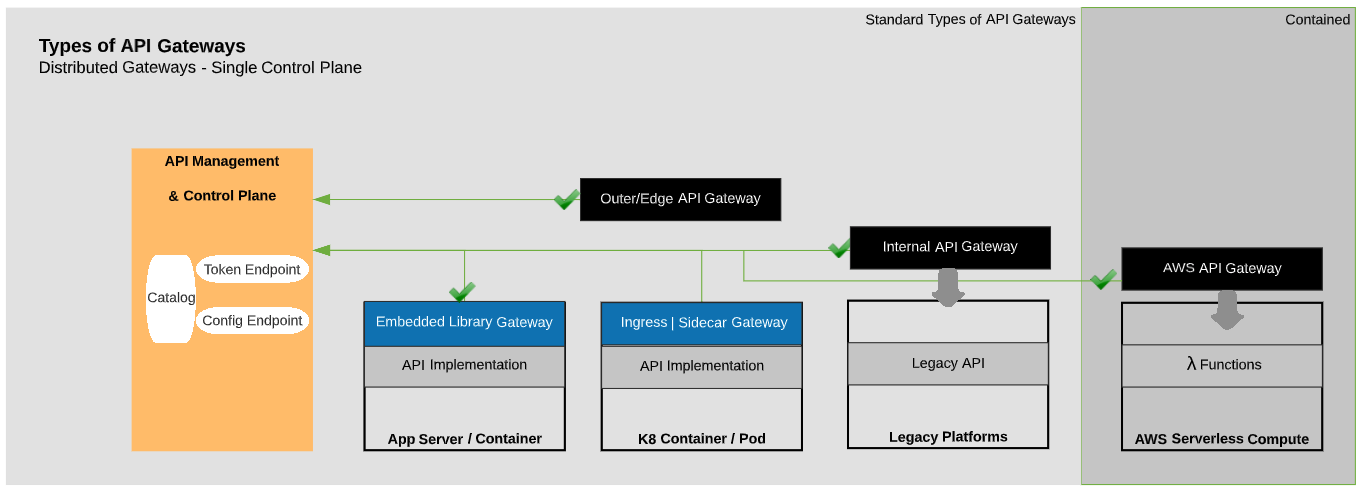
This is the same high level reference architecture model reflecting a viewpoint of 3 different types of runtime/data planes.



**Comments on the HL Reference Architecture Model**

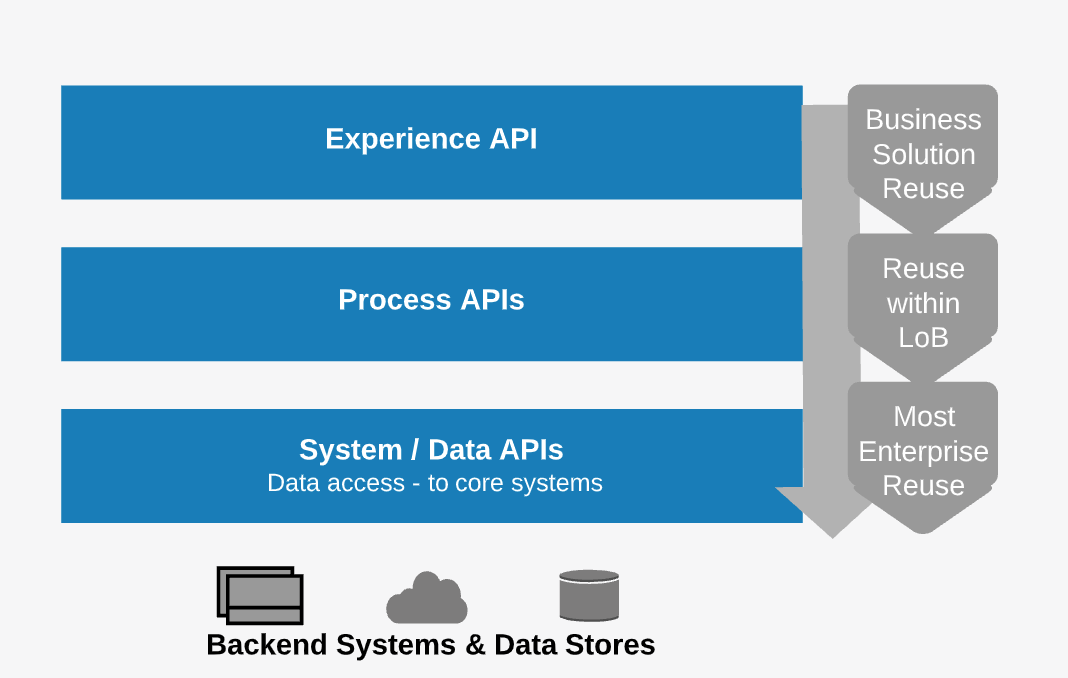
**Types of API Gateways**

From the Ref Arch Model one can infer these types of API Gateways.. All Gateways are distributed in the data plane while connecting to a common management/control plane.



It's worth noting that with this architecture model an important principle is to implement security policy enforcement as close as possible to the API workload endpoint.

**Layered API Taxonomy**



This taxonomy helps with driving a **value-driven Reuse**. Most enterprise valuable API Reuse happens at the System/Data API layer, so reuse of certain capability enabled by lower layers APIs is even multiplied under the cover each time applications are added at the surface at any higher up layers

This taxonomy also helps with one of the most frequently asked question: **What if I have an API that I would like to expose externally?** to address this scenario, once reaching a good understanding and applying this reference architecture, the question becomes: ***Does my Process API need an Experience API? how the Experience API would be aligned with the target channel?***

More details about that in this next comment...

**Layered API Implementation**

