Service Orchestrator Architecture Enhancement - Proposal

Seshu Kumar M

So… Whats Next So…

- Dynamism
  - Customization
- Orchestration next steps
  - CNF support
  - TOSCA
- Plug and Play
  - Both Yellow and Green Field adoption

Further Topics on this…

13th Jan

11:45		for VNF Hack Track participants (1hr)			**ONAP Service Orchestrator** Roadmap @ Seshu Kumar M

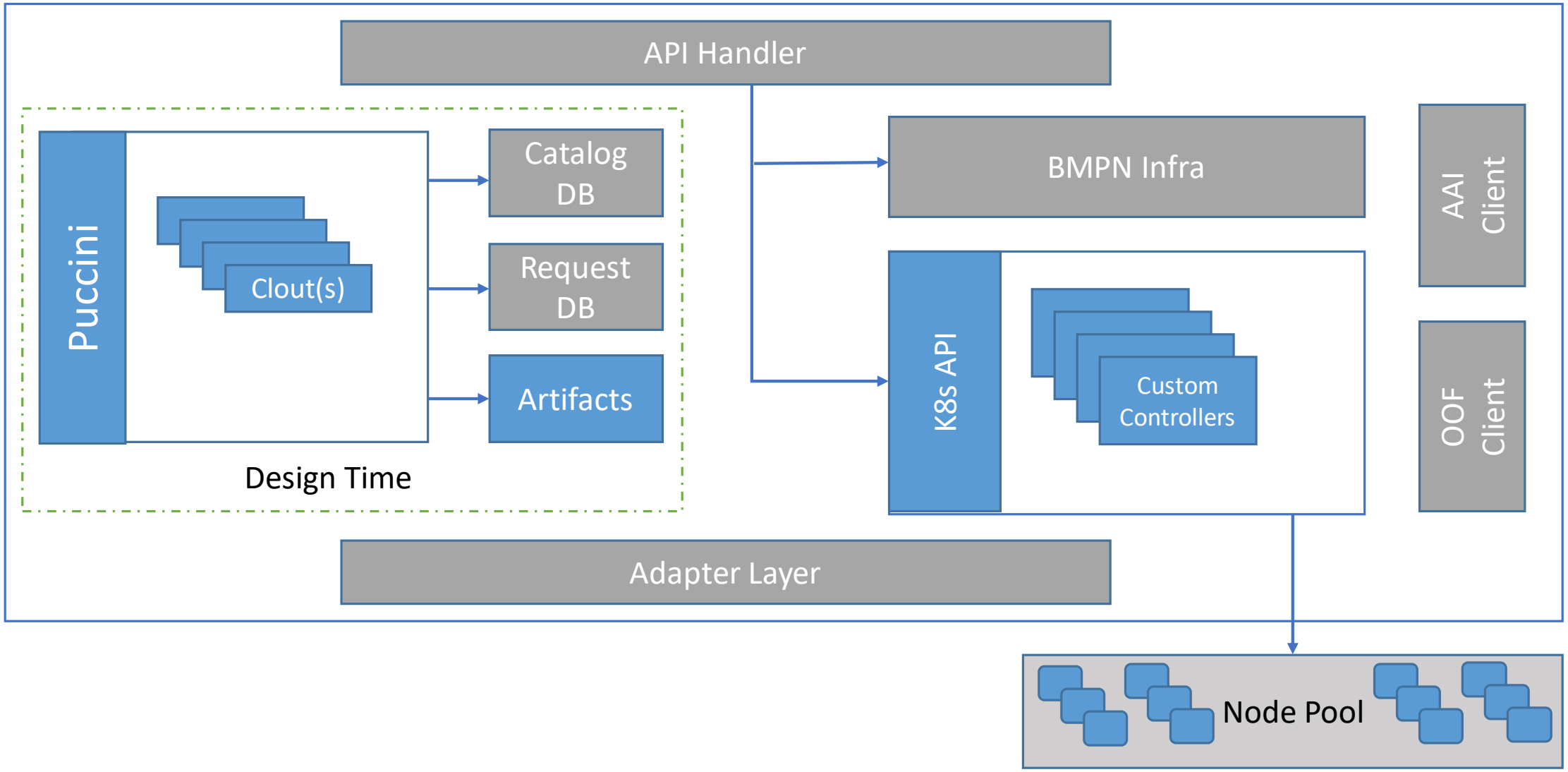
15th Jan

14:00	**Joint CNTT/OPNFV: OPNFV TSC 2.0** Discussion on the future directions of OPNFV with the CNTT @ Jim Baker	**Hack/Plugfest** • Plugfest Track Details	**Attend the Joint CNTT/OPNFV in South Hall 2**	**ONAP/ETSI Alignment** - Support Status and Plan @ Byung-Woo Jun @ Fernando Oliveira @ Miroslaw Medrek

14th Jan

15:00	how to govern it, work with CNTT, CNCF, etc) • **When** (what are the timelines, regular mtg cadence/time) Facilitators: @ Lincoln Lavoie @ Marc Price @ Rabi Abdel @ William Diego	OPNFV TSC meeting placeholder - remote access @ Jim Baker @ Al Morton Agenda (starts at 1500hrs local time or 1400 UTC)	EUAG f2f meeting (starts at 1500hrs local time or 1400 UTC) Zoom Bridge	**Modeling subcommittee** @ Hui Deng @ Andy Mayer (starts at 1500hrs local time or 1400 UTC)		**Plug and Play in ONAP** - Learnings of a developer @ Seshu Kumar M @ Isaac Manuel Raj

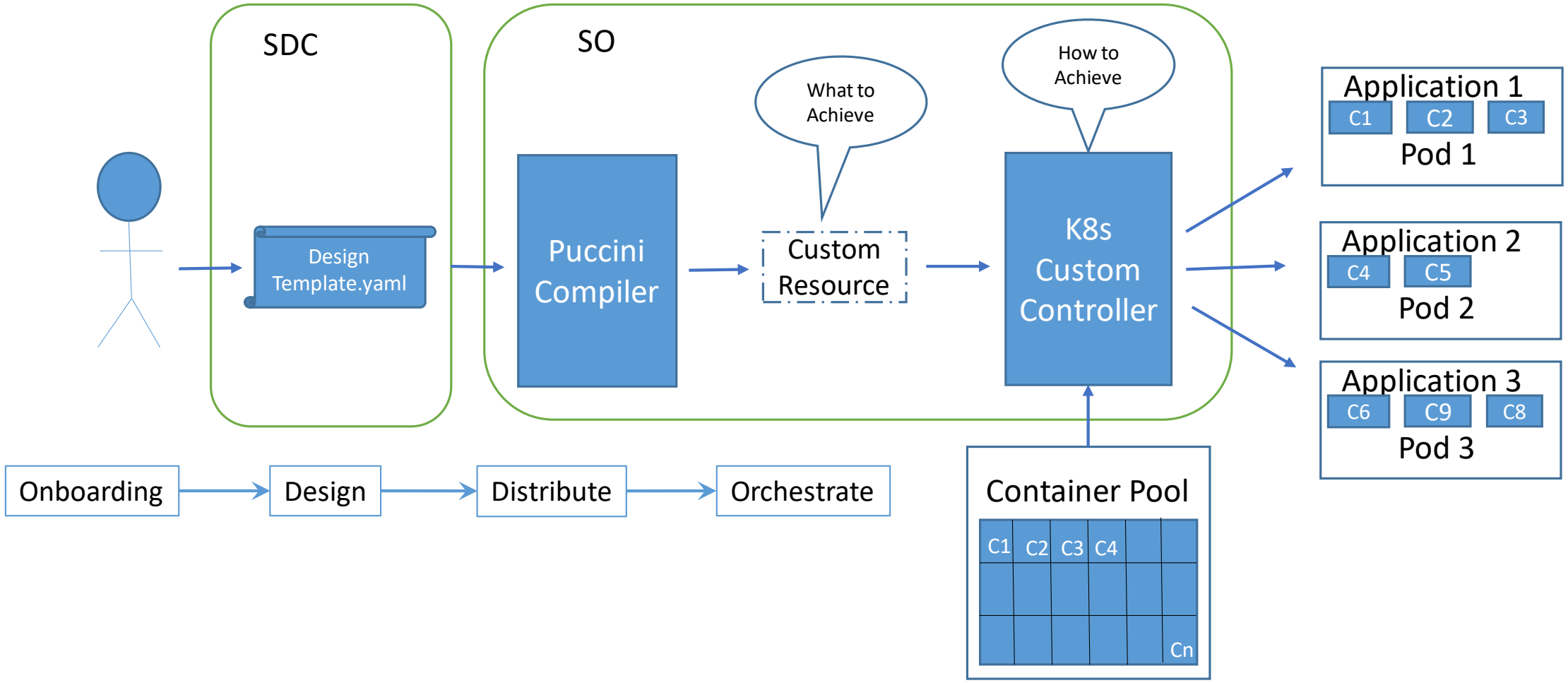
SO Proposed Architecture



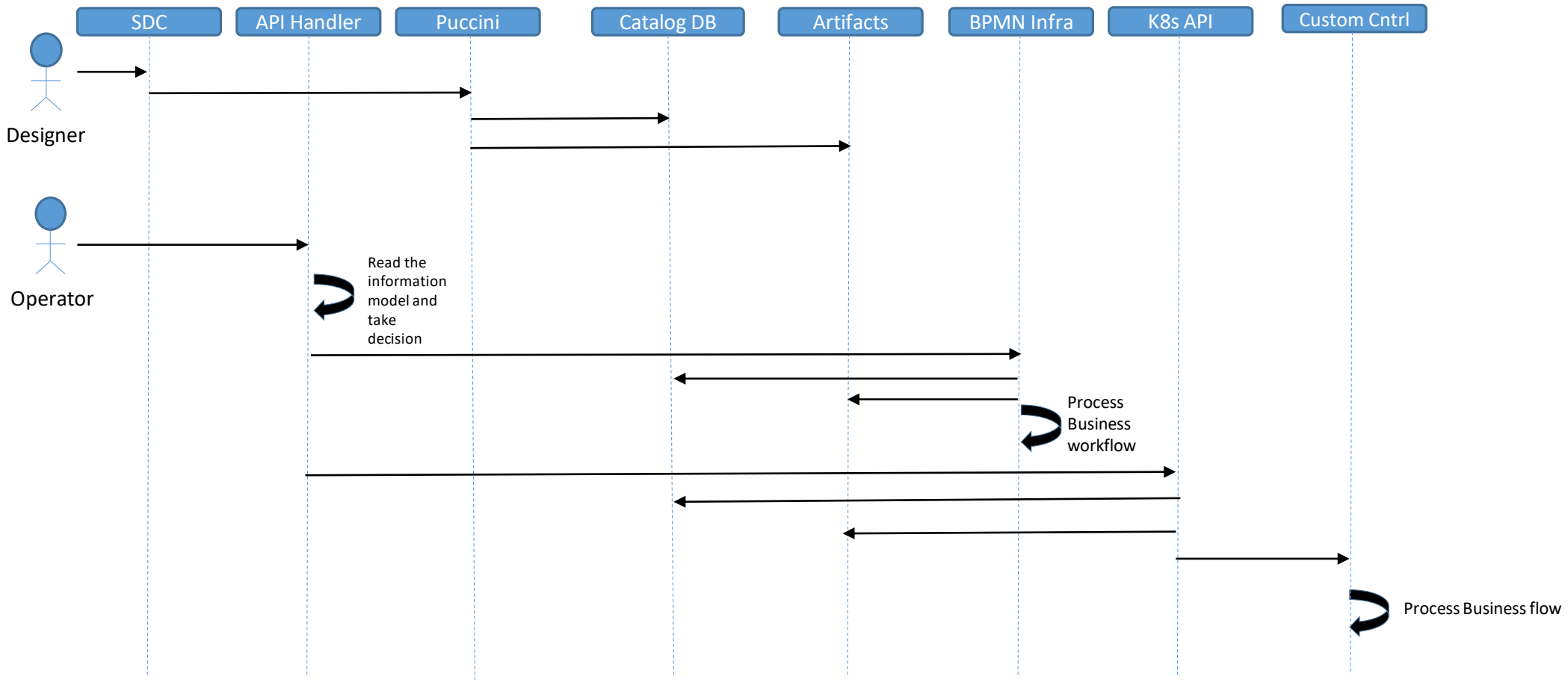
Key additions to the Architecture

- Puccini
  - A TOSCA compiler that parses a given TOSCA service template and compiles it to Clout (Cloud Topology)
  - Puccini-tosca comes with TOSCA profiles for the Kubernetes and OpenStack cloud infrastructures, as well as BPMN processes.
  - Profiles include node, capability, relationship, policy, and other types that would work with any TOSCA-compliant product.
- Artifacts
  - Constitutes the design time entities that are onboarded to SDC and distributed to SO.
  - These could include the configurations, custom workflows designed, custom resource definitions, etc…
- K8s API
  - This constitutes of 2 key components Custom resources and Custom controllers.
- Custom resources
  - A *resource* is an endpoint in the K8s that stores a collection of API objects of a certain kind.
  - Custom resource is an extension of the Kubernetes API that is not necessarily available in a default Kubernetes installation.
  - It represents a customization of a particular Kubernetes installation and hence help in making Kubernetes more modular.
- Custom controllers
  - Custom resources let you store and retrieve structured data. When you combine a custom resource with a custom controller, custom resources provide a true declarative API.
  - Custom controllers interprets the structured data as a record of the user's desired state, and continually maintains this state.
  - CCs can work with any kind of resource, but they are especially effective when combined with custom resources.

Typical Functional Flow



Typical Functional Flow



Advantages

- The new architecture would leverage the existing ONAP SO functionality to even orchestrate the CNFs
- It brings in the advantages of a customization of resources aka Network functions and provides the bundles of advantages of K8s included with it.