



**ONAP**

OPEN NETWORK AUTOMATION PLATFORM

# ONAP Architecture Developer Conference

Chris Donley

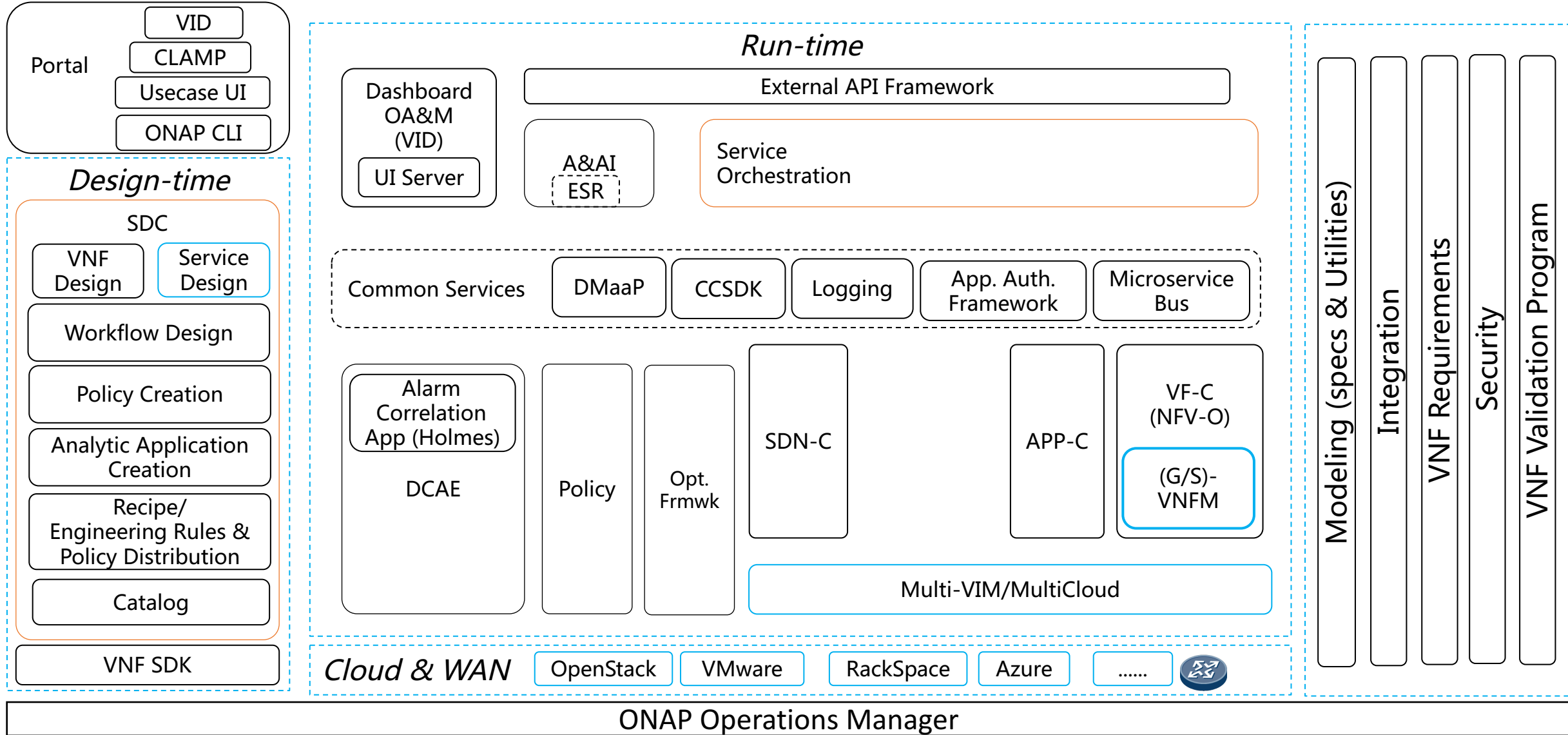
# Architecture Subcommittee: Charter

- The architecture subcommittee is responsible for developing and maintaining a functional ONAP architecture. This functional architecture helps inform relationships and interaction between functional modules, which may include high level information flows between the modules supporting the use case(s) driving each release. It also helps the community with project proposals by clarifying the new project relationship with existing components.
- The architecture subcommittee will not make decisions regarding internal functioning of projects.
- The architecture subcommittee is advisory by nature, and not authoritative. It may provide advice to projects and to the TSC, such as by providing a forum to help resolve architectural questions that may arise.
- The architecture subcommittee operates on a rough consensus basis. If the subcommittee is unable to reach consensus on what advice to offer, the subcommittee will refer the matter to the TSC or inform the project that advice cannot be rendered.
- The architecture subcommittee will consult with Projects to help drive alignment between components and with the functional architecture.

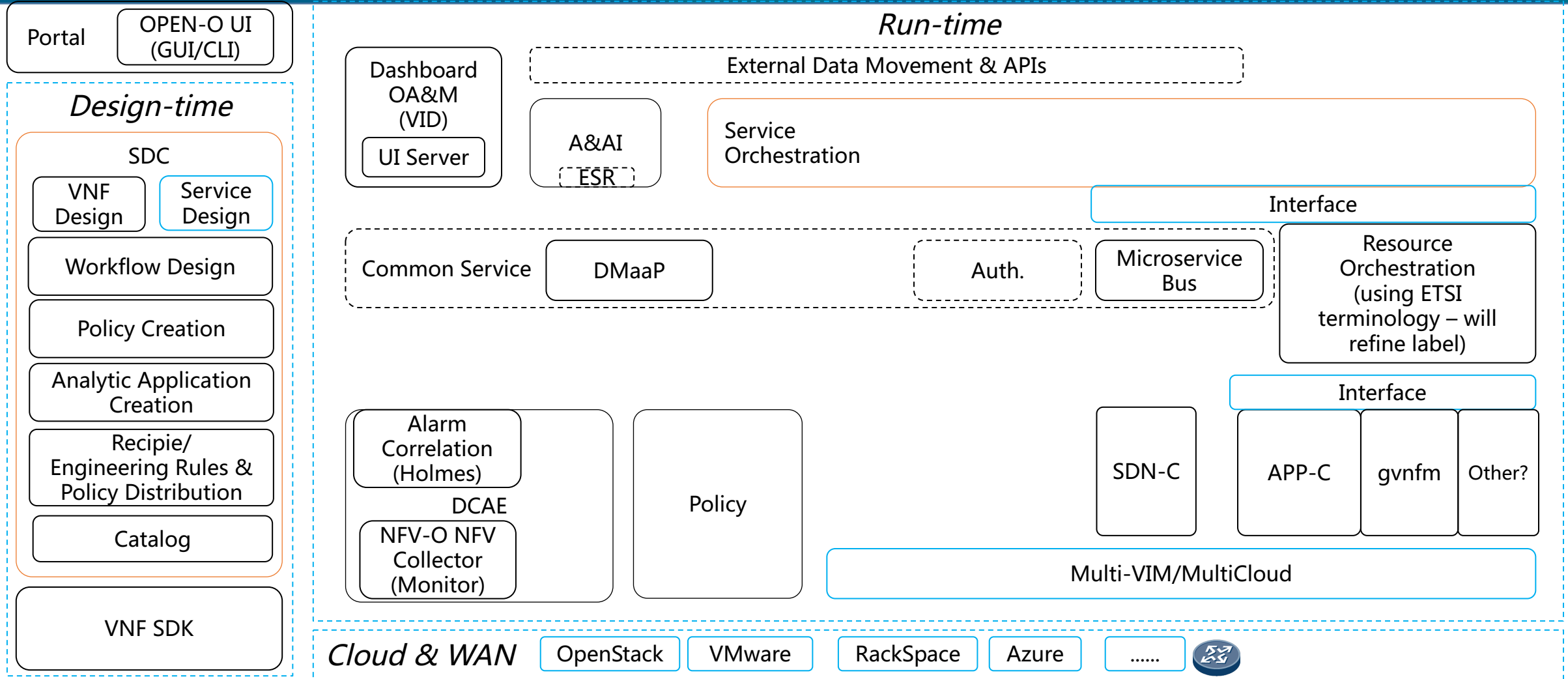
# Architecture Subcommittee Goals

- Develop and maintain functional architecture and principles to facilitate interaction between projects
- Provide advice and help resolve architecture issues that may arise
- Conduct mid-release walkthroughs to help facilitate alignment between projects
- Define information flow between components

# Architecture Baseline for R1 (with projects)



# FUNCTIONAL Architecture for R2+



Consensus on layered architecture. This picture does not imply project structure.

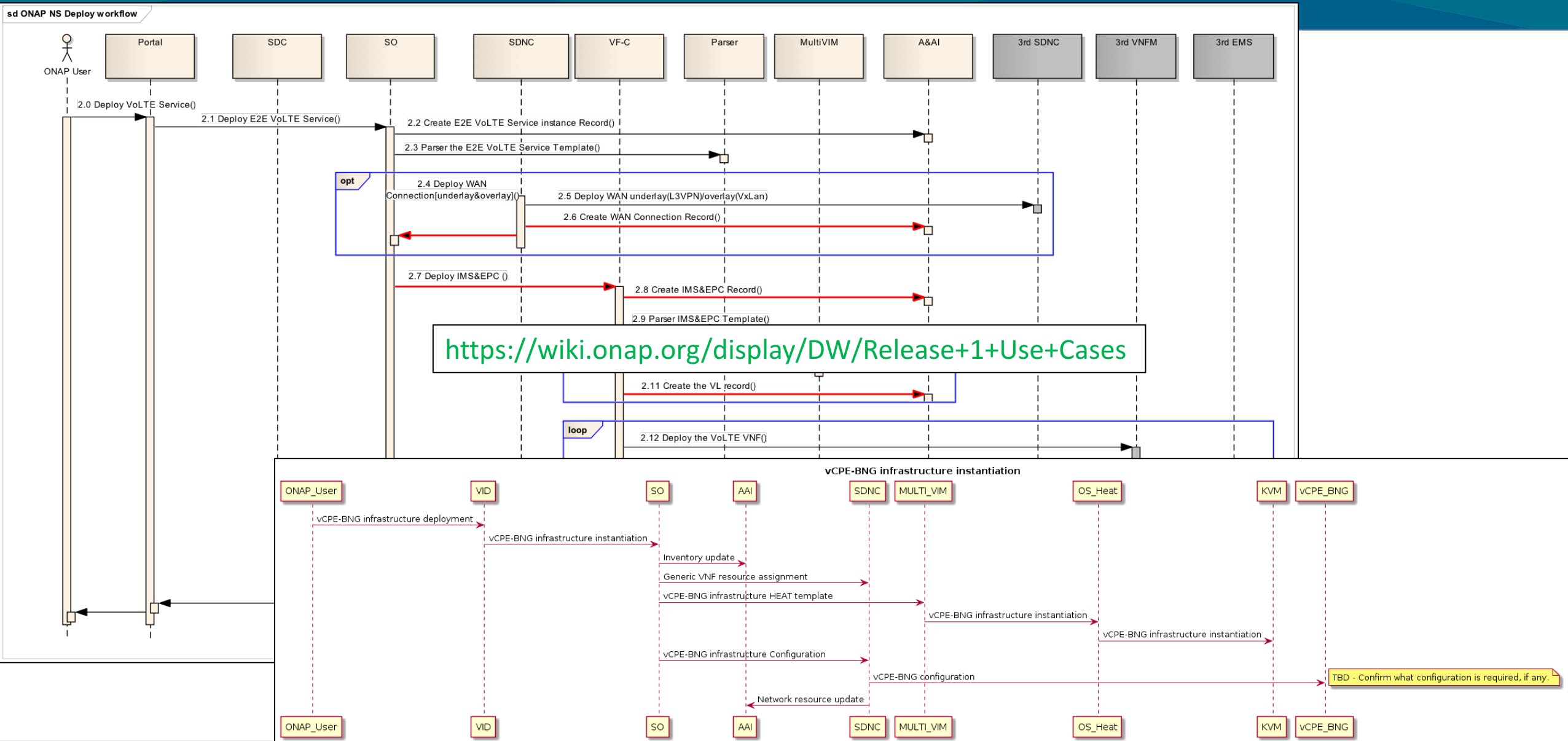
# Steps towards R2+ architecture

- ✓ SO and VFC meeting to identify R1 'baby steps' (if any)
- Seshu took action item to set up meetings before we open R2 planning to address:
  1. Map projects into functional architecture
  2. Define interfaces between layers
  3. Map existing code into (1) and (2)
    - (project-level discussion, not ARC committee)

# Mid-release walk-throughs

- Goals:
  - Check project interface/API alignment prior to API Freeze milestone
  - Identify any misalignment or open issues
- Methodology
  - Review dependency graph and use case flows
  - Identify APIs/interfaces per project
  - Check alignment between adjacent projects
  - Target 10 minutes per project

# Use Case Flows





# Information to review

- Flow
  - API name/URI
  - Type
  - Direction (incoming/outgoing)
  - Adjacent module(s)
  - Information expected
  - Information provided