



# AAI Model Visualization R2 (Beijing)

Jimmy Forsyth

# AAI Model Visualization

- AAI's data model is complex
- Parent/Child edges: nesting of node types and re-use of node types means a node type can belong to more than one parent node type
- Cousin edges: various types can be related to each other in multiple ways
- Edge rules JSON file hard to read
- Model visualization intends to expose the model to designers or operators to display graphically the set of node types and the possible relationships between them

# AAI Model Visualization - GraphGraph

- GraphGraph is a set of tools created by Jane Threefoot from AT&T that ingests the A&AI schema files in order to provide dynamic documentation
- Three microservices:
  - Core
  - Web UI
  - Chatbot
- Any of these microservices could be repurposed to serve as dynamic documentation for another system/schema/etc

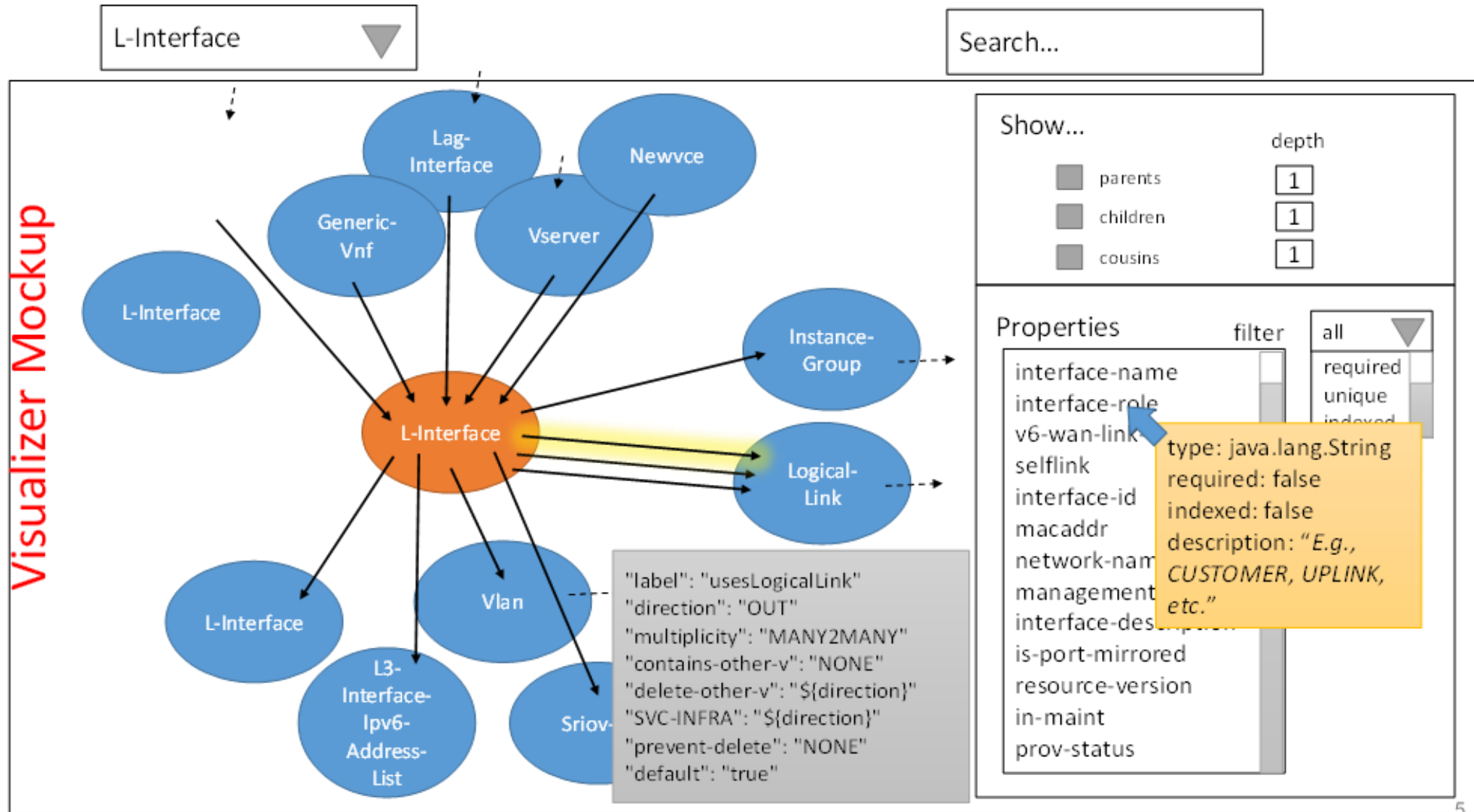
# AAI - GraphGraph

- Ingests schema files (node oxm and edge rules json) to build an in-memory sample graph with one example of every node and edge
- Provides REST API for querying this sample graph

# AAI - GraphGraph

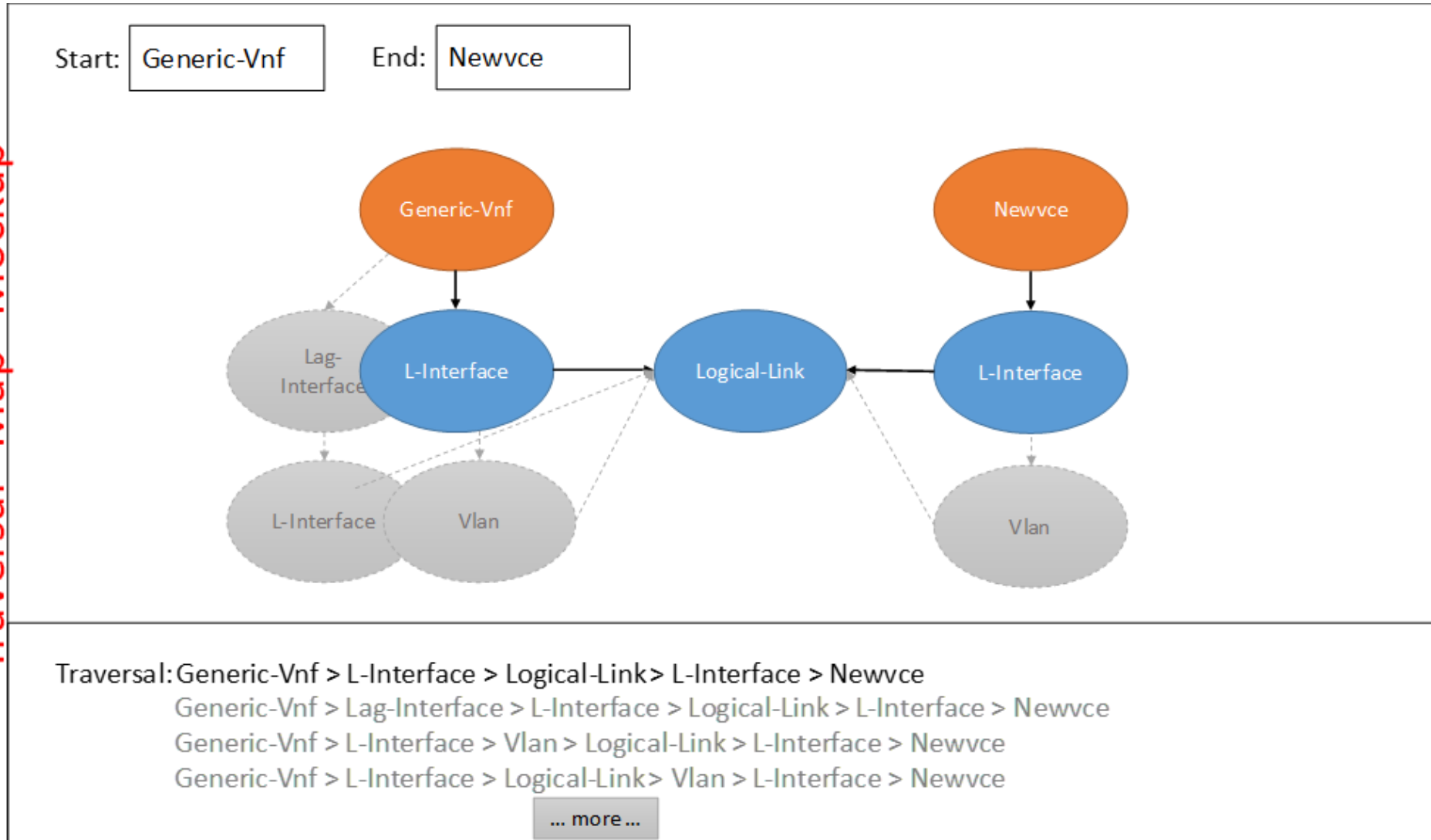
- Uses Core REST API to visualize the schema
- Provides “Google Maps”-esque traversal building
  - Provide start and end nodes, optional pass-through nodes
  - GGUI provides shortest possible traversals that satisfy given constraints (or an error if there is no valid path)
  - Allows users to drag’n’drop traversal to other related nodes to build alternate valid traversals
- Could potentially leverage Amdocs’ A&AI UI work

# AAI – GraphGraph – visualizer mockup



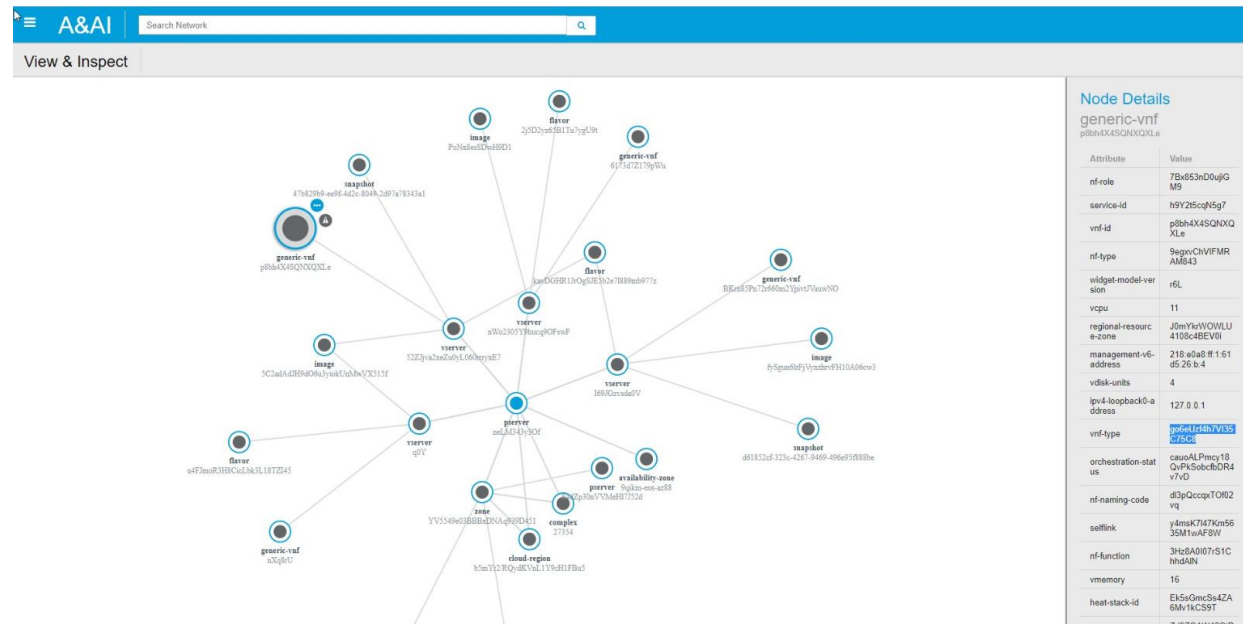
# AAI – GraphGraph -

Traversal “Map” Mockup



# AAI – Sparky UI

- AAI Provides a UI front end nick-named “Sparky” which allows users to walk the graph of actual instance objects (generic-vnfs, service-instances, pnfs, I3-networks, etc)
- Proposal is to augment sparky to use GraphGraph as its backend to provide visualization of the AAI model





# AAI – Other Model Visualization Efforts

- Pavel and Adrian from Orange have put together a POC visualizer of the AAI schema and edge rules:

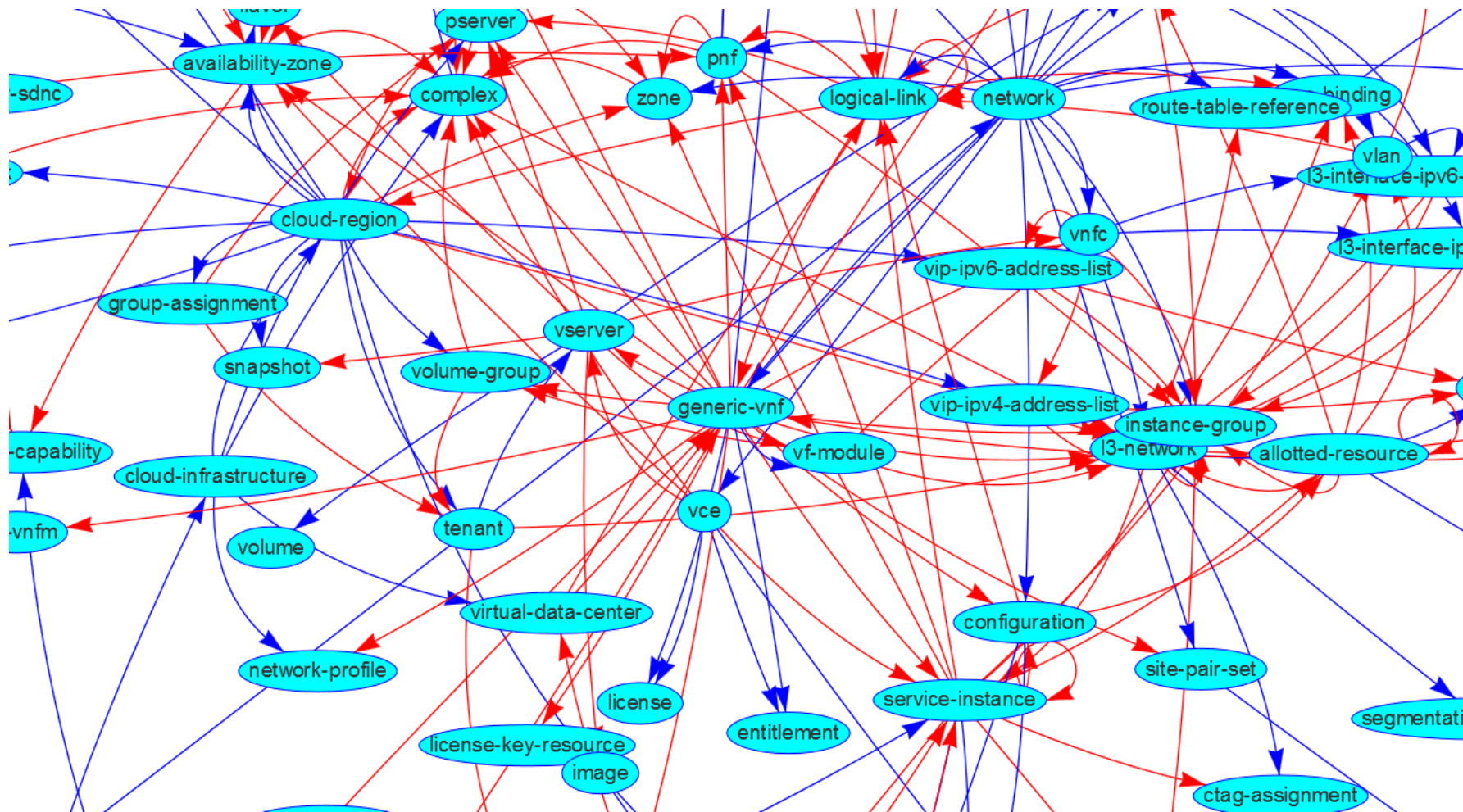


# AAI – Other Visualization Efforts

- Told you: it is complicated...

# AAI – Other Visualization Efforts

- Zoomed in view (you can try yourself @ <https://hotlib.github.io/>)





**ONAP**

OPEN NETWORK AUTOMATION PLATFORM

Thank You!