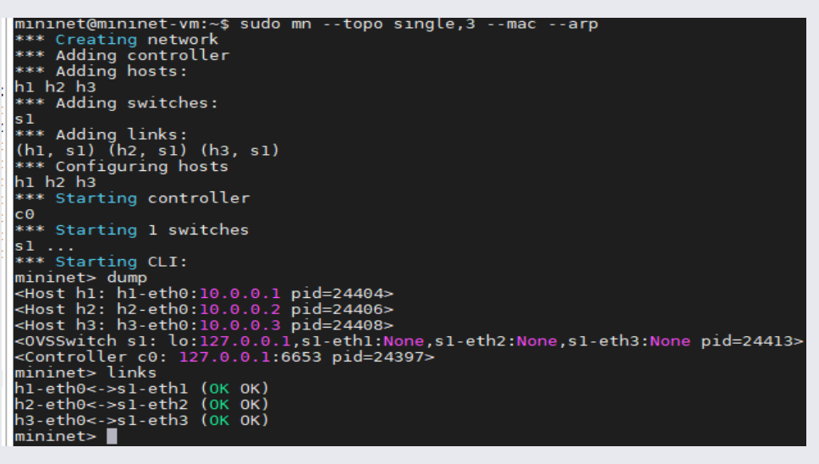
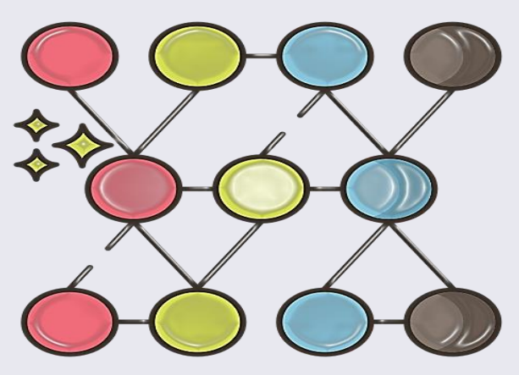
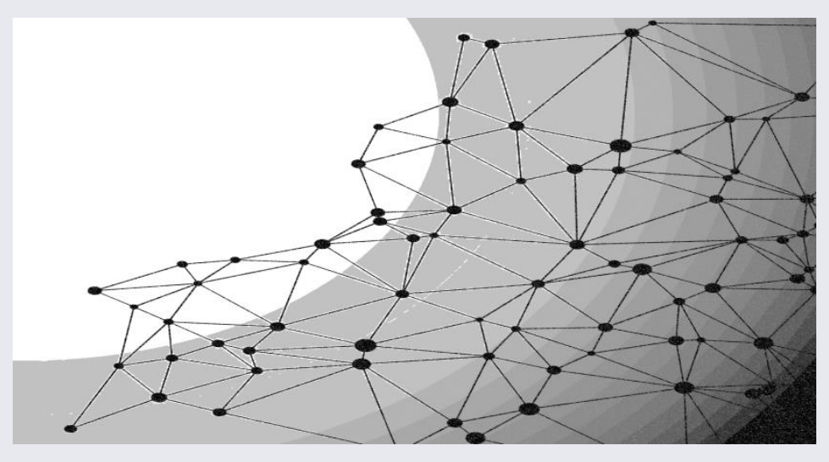
V-SDN : Visualizer for Software Defined Network

Marius Faust, Manjiri Birajdar, AOC-4

TECHNISCHE UNIVERSITÄT DARMSTADT

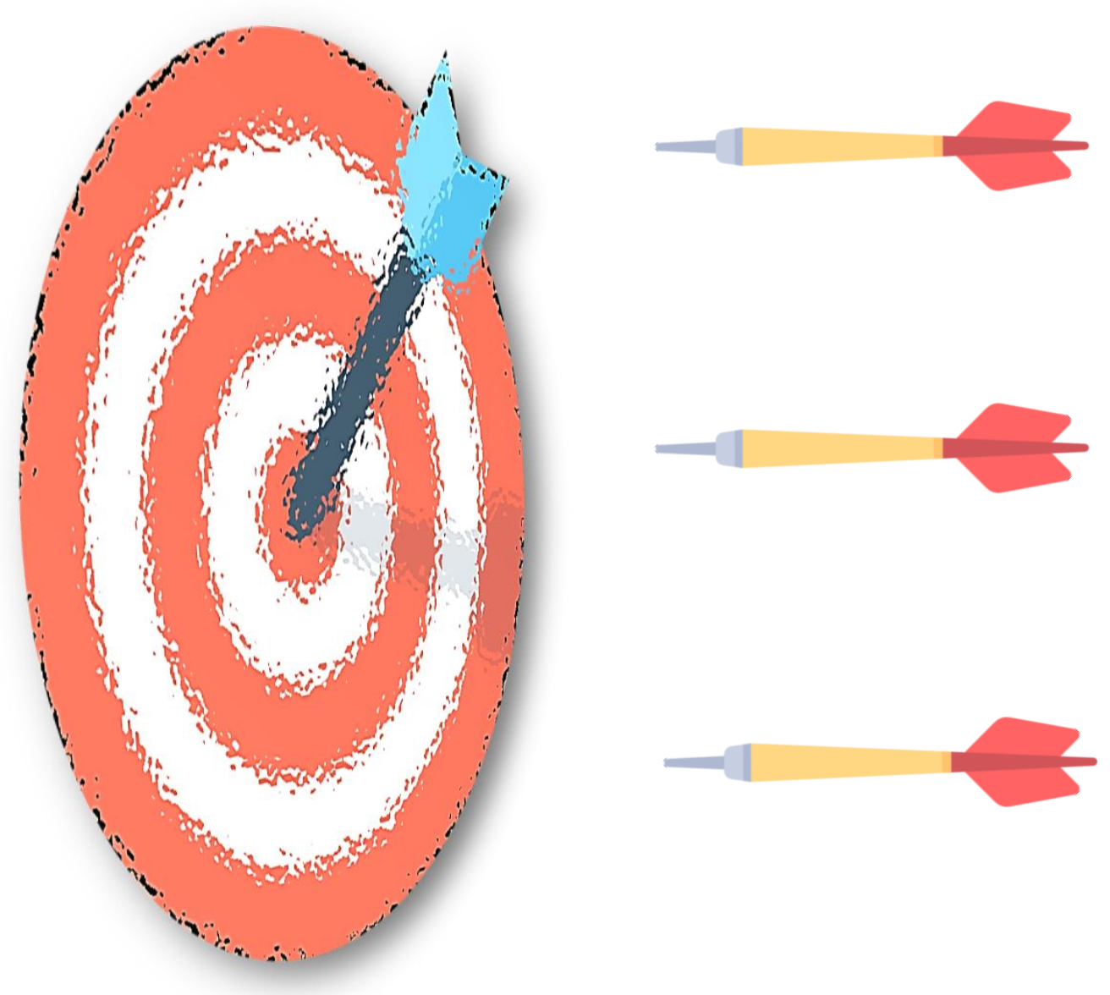
Motivation

Why visualize SDN ?

		
Build User Friendly UI	**Node-link structures are more intuitive**	**Easy to grasp complex coherences**
- CLI, text-only interface, difficult to read - No graphical user interface (GUI) to navigate through a system	- Understanding event occurrences - Visualize entities communication	- Simplify complexity - Look at the context and understand details - Identify different networks and its nodes-links

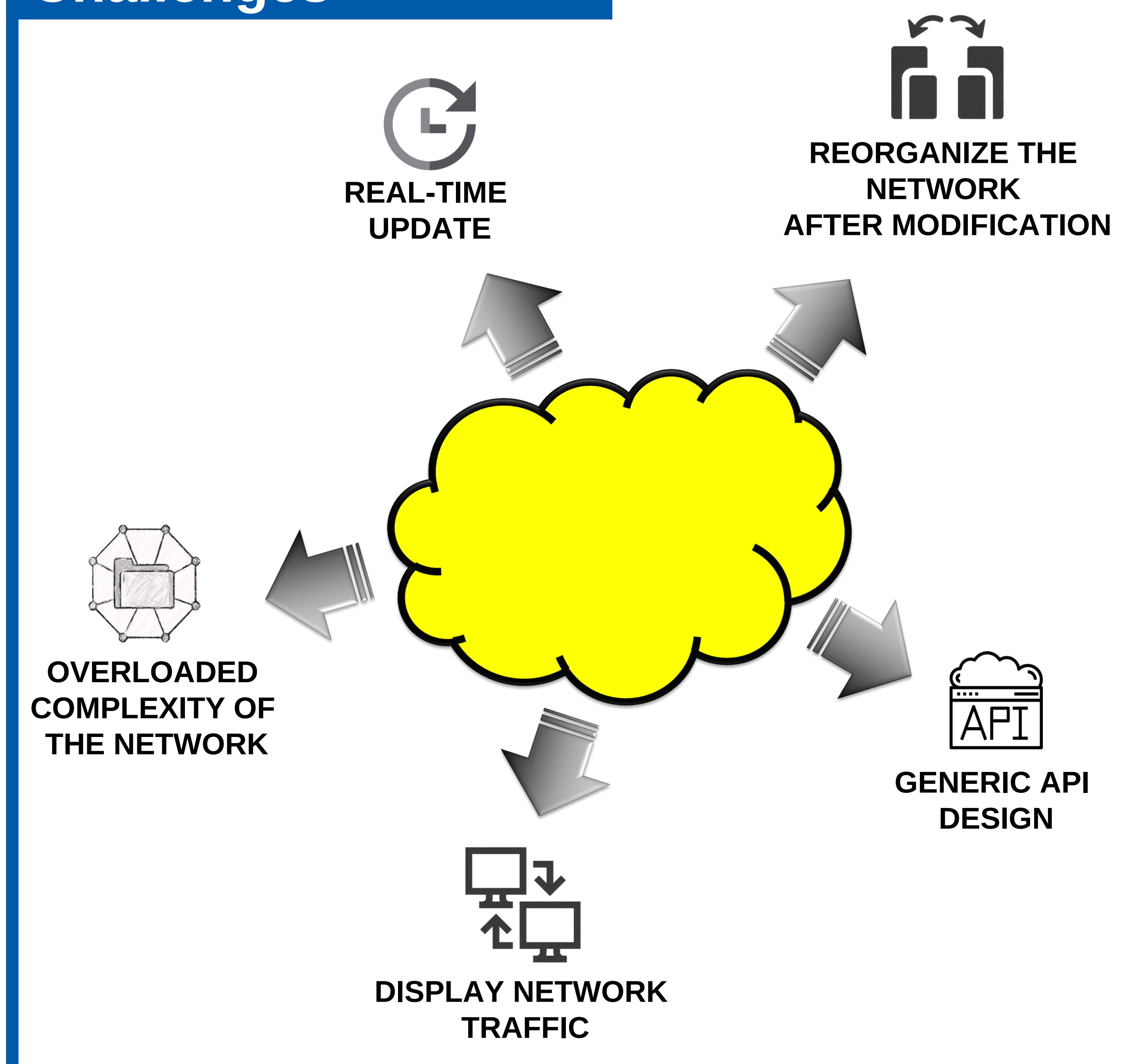
Goals

Development of ...



- Visualizer for SDN (Display Network Map)
- Generic APIs with extensible Functionality e.g. addSwitch(…), addHost(…), createNetwork(…)
- Provide accessibility to various programs like JAVA SDN Controller, python scripts

Challenges

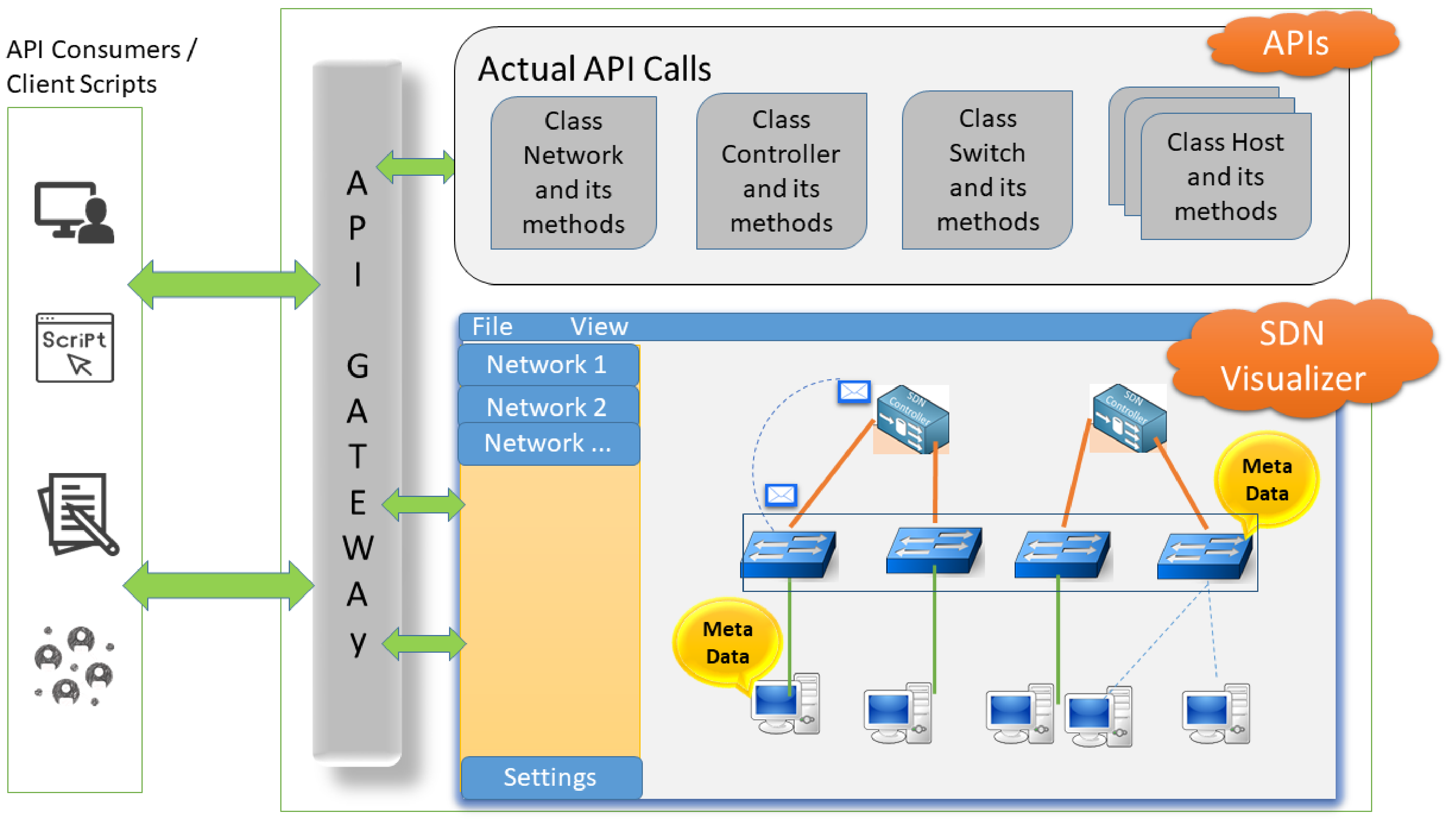


REAL-TIME UPDATE
REORGANIZE THE NETWORK AFTER MODIFICATION
OVERLOADED COMPLEXITY OF THE NETWORK
GENERIC API DESIGN
DISPLAY NETWORK TRAFFIC

Exsisting Approaches

Approach	Strength	Weakness
HPE Virtual Application Networks (VAN) SDN Controller / Aruba Network Visualizer App.	➢ Provide dynamic traffic capture with real-time ➢ Detailed network monitoring	➢ This product has been discontinued and is not for sale by HPE in any region. ➢ Dependency on Server: Aruba VAN SDN Controller software
SPEAR : SDN Narmox Spear (application enhances the functionality of HPE VAN SDN Controller)	➢ Complete tool for SDN network administration ➢ Manage network traffic ➢ Shows graphical overview of network‘s current state	➢ Dependency on Server: Aruba VAN SDN Controller software ➢ Not open source: Trial Licence for 14 days and further requires Commercial Licence
Visual Network Description (VND)	➢ Authoring of SDN Network Scenarios via GUI. ➢ Automatic creation of Mininet Scripts ➢ Automatic creation of Openflow Controllers Scripts	➢ Drag and drop components ➢ Cannot be accessed by code
Mininet GUIs: consoles.py and miniedit.py	➢ GUI for creating a Mininet ➢ Display details of each node in the network	➢ No runtime visualization for the network and its architecture ➢ Cannot change the number of nodes dynamically
Mininet Editor	➢ Visualizes Mininet ➢ Apply changes via drag and drop ➢ Export topology and create new Mininet	➢ Cannot be accessed by code ➢ Dependency on server/website ➢ No runtime visualization

Our Solution



API Consumers / Client Scripts
Script
API GATEWAY
Actual API Calls
APIs
Class Network and its methods
Class Controller and its methods
Class Switch and its methods
Class Host and its methods
File
View
Network 1
Network 2
Network ...
Settings
SDN Visualizer
SDN Controller
SDN Controller
Meta Data
Meta Data

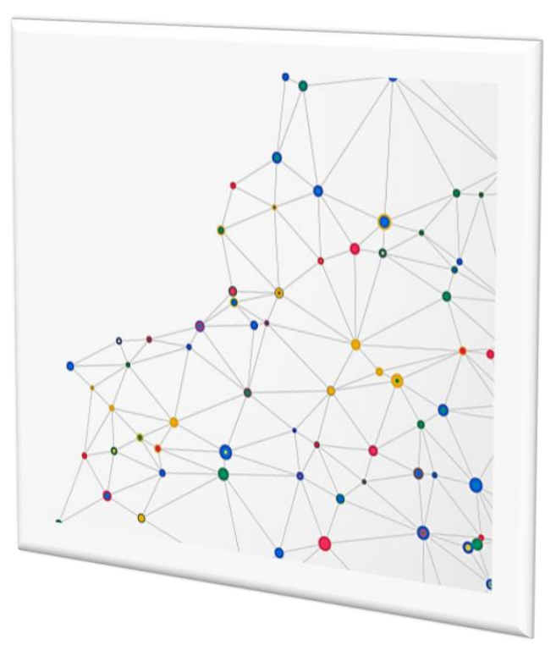
API Implementation:

- Use **Python** as base to implement an APIs with the desired function-calls

  e.g. sendMessage(to,from), addSwitch(…), addHost(…), addLink(..), onClick(…) create and manipulate the necessary objects in Python
- Use **TCP** as a protocol and **JSON** format for message-encoding

GUI :

- Displays the data in a Simple GUI using **D3js** or **gephi**
- Node-Links with different colors and thickness
- Update GUI according to the Python-objects and show the **network topology/map**
- Display **meta-data** ‘onClick’ event on the GUI-objects



Future Steps

- ➢ Improve overall functionality by adding more APIs
- ➢ Add Real-time update feature for complete network
- ➢ Add detailed network moitoring

References and acknowledgements: ***Icon made by Freepik from*** *www.flaticon.com*, *http://spear.narmox.com/*, https://github.com/ramonfontes/vnd, https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-c03967699