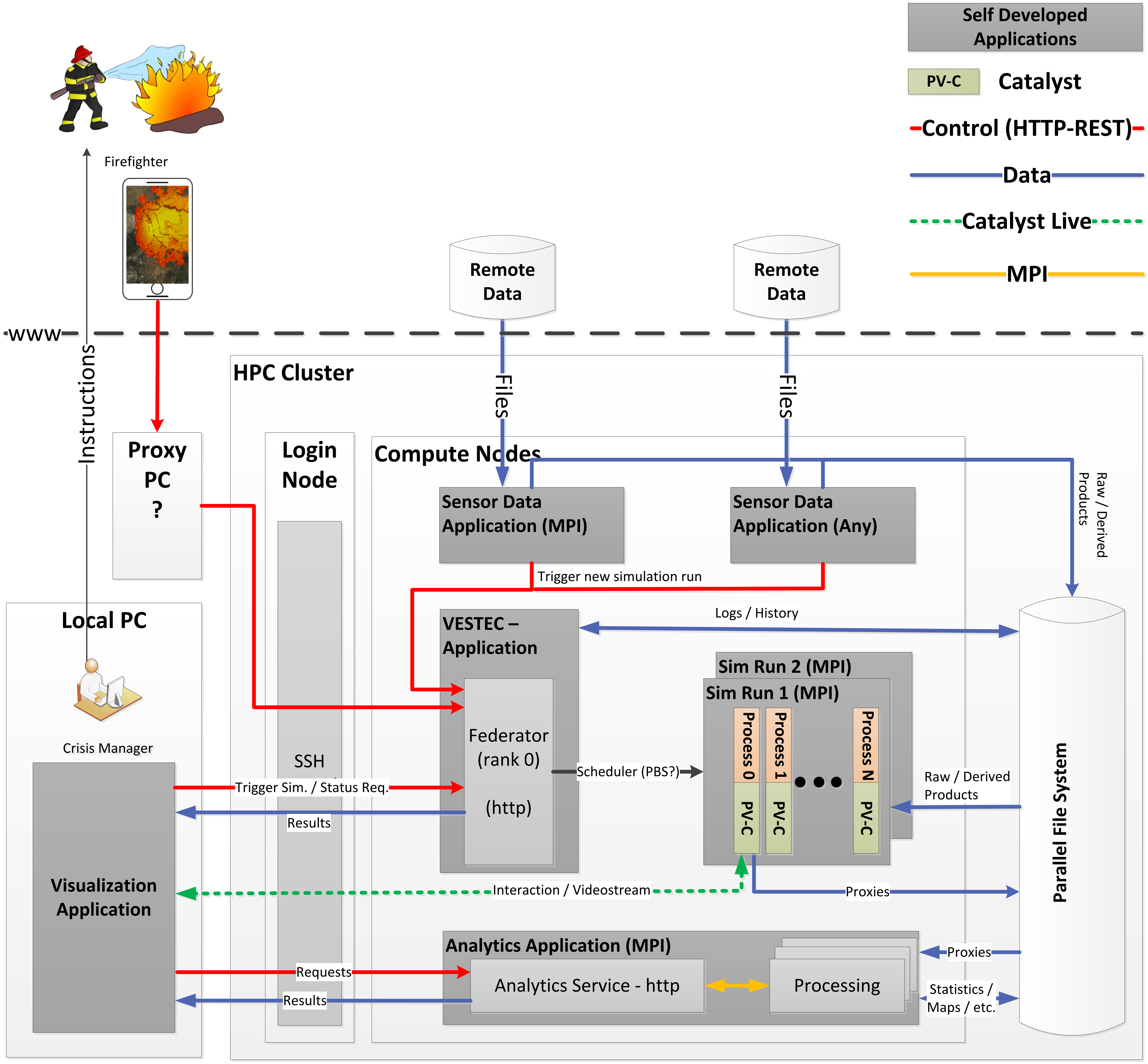


Remarks:

- Connections could be established via SSH tunnels

Open questions:

- How or good ways to integrate / register
  - topological methods (Catalyst)
  - Metrics to analytics applications
- Is a proxy required? And when how does it get the comute node information?

VESTEC – Application

- configured via config file
- Path for storing logs and proxies

Analytics – Application

- Define available metrics to compute statistics for different proxy types
- Path to store statistics

HTTP / REST

- Pistache REST API (C++)?

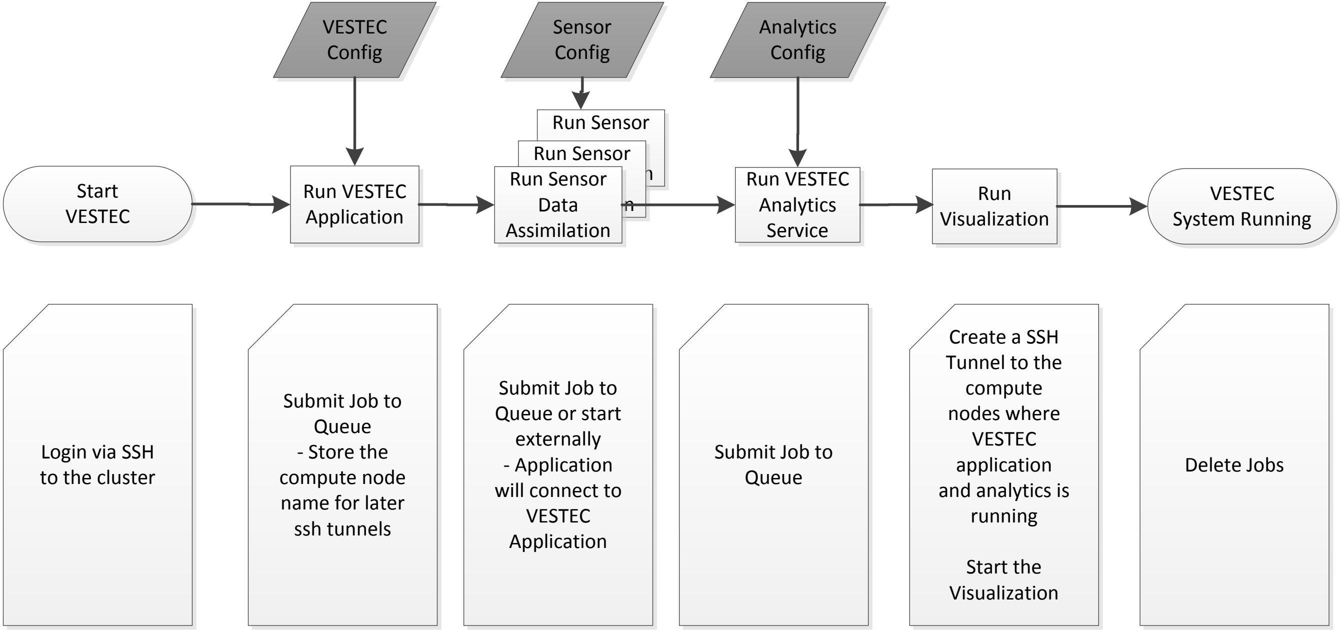
Possible Messages Federator

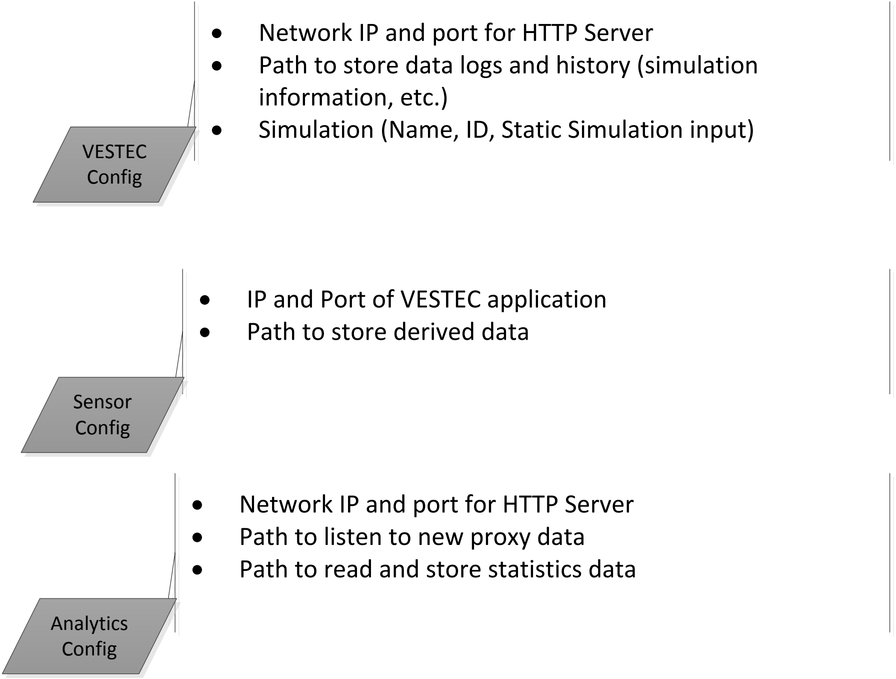
- GET – VESTEC System Status – Return Number of Simulations, Busy Processes, Connected Sensor Sources, etc.
- POST – Trigger new Simulation(SimID, Sim Params, Serialized Sensordata?, Path to Sensordata or derived products) - Return ComputeNode name
- POST – Re-Run Simulation (SimID) – Return ComputeNode

Possible Messages Analytics

- GET – getSimulationProxy (SimID) – Return VTK + Metadata
- GET – getStatistics () – Return VTK + Metadata

- **Proxies (Data)**
  - Meta data + VTK Dataset
  - Meta data: list of metrics to compute, sim id, etc.


VESTEC Config
Sensor Config
Analytics Config
Start VESTEC
Run VESTEC Application
Run Sensor
Run Sensor
Run Sensor Data Assimilation
Run VESTEC Analytics Service
Run Visualization
VESTEC System Running
Login via SSH to the cluster
Submit Job to Queue - Store the compute node name for later ssh tunnels
Submit Job to Queue or start externally - Application will connect to VESTEC Application
Submit Job to Queue
Create a SSH Tunnel to the compute nodes where VESTEC application and analytics is running
Start the Visualization
Delete Jobs



VESTEC Config

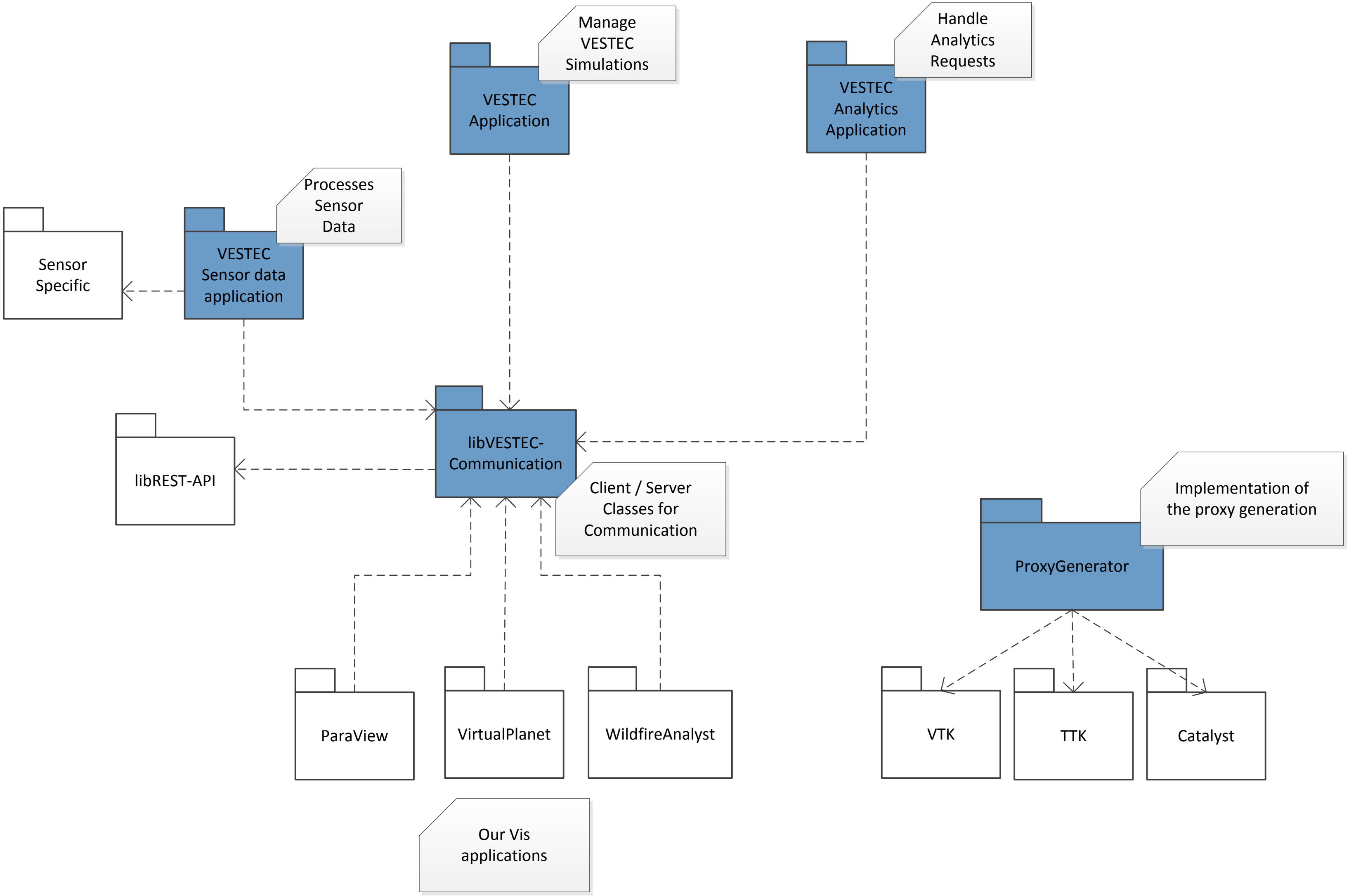
- Network IP and port for HTTP Server
- Path to store data logs and history (simulation information, etc.)
- Simulation (Name, ID, Static Simulation input)

Sensor Config

- IP and Port of VESTEC application
- Path to store derived data

Analytics Config

- Network IP and port for HTTP Server
- Path to listen to new proxy data
- Path to read and store statistics data


Manage VESTEC Simulations
VESTEC Application
Handle Analytics Requests
VESTEC Analytics Application
Processes Sensor Data
Sensor Specific
VESTEC Sensor data application
libVESTEC-Communication
libREST-API
Client / Server Classes for Communication
Implementation of the proxy generation
ProxyGenerator
ParaView
VirtualPlanet
WildfireAnalyst
VTK
TTK
Catalyst
Our Vis applications