

# Using SCiMMA architecture for SNEWS server

Amanda Depoian & Skylar Xu

[adepoian@purdue.edu](mailto:adepoian@purdue.edu)

[yx48@rice.edu](mailto:yx48@rice.edu)

SNEWS Collaboration Meeting - June 19, 2020

# SCiMMA

## Scalable Cyberinfrastructure to support Multi-Messenger Astrophysics

- NSF project developing tools for inter-observatory communication and a network for alerts.
- Their long term support plan is an intended 5 year institute starting after this 2 year period.
- SCiMMA is very keen on collaborating and supporting SNEWS collaboration.

# Hopskotch

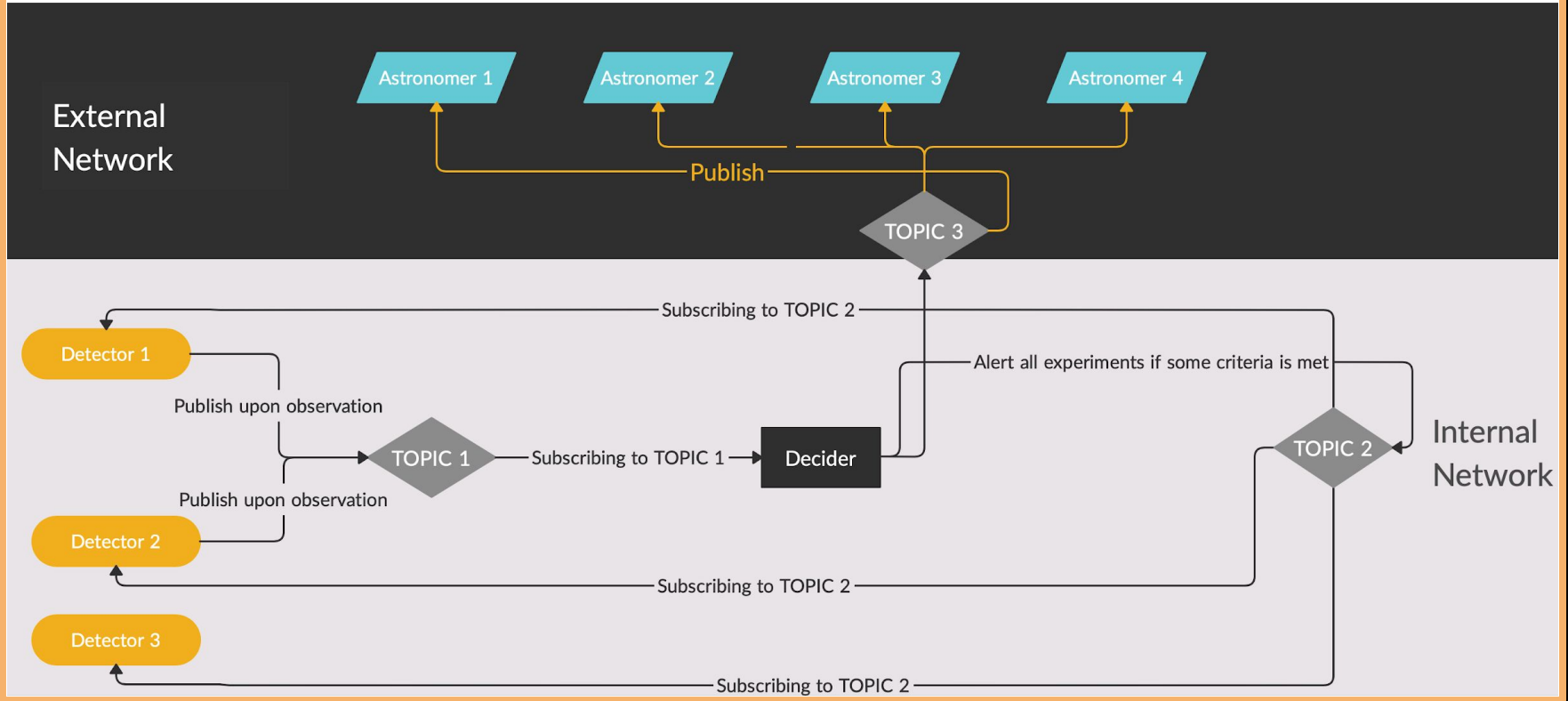
- Hopskotch is a publish-subscribe messaging system implemented based on Apache Kafka (stream-processing software platform).
- SCiMMA deploy their server on AWS (Amazon Web Services); The client is currently the Python package “hop-client”.
- Clients can publish/subscribe messages using command lines such as:
  - **hop publish kafka://SERVER/TOPIC -F config.conf example.gcn3**
  - **hop subscribe kafka://SERVER/TOPIC -F config.conf -e**

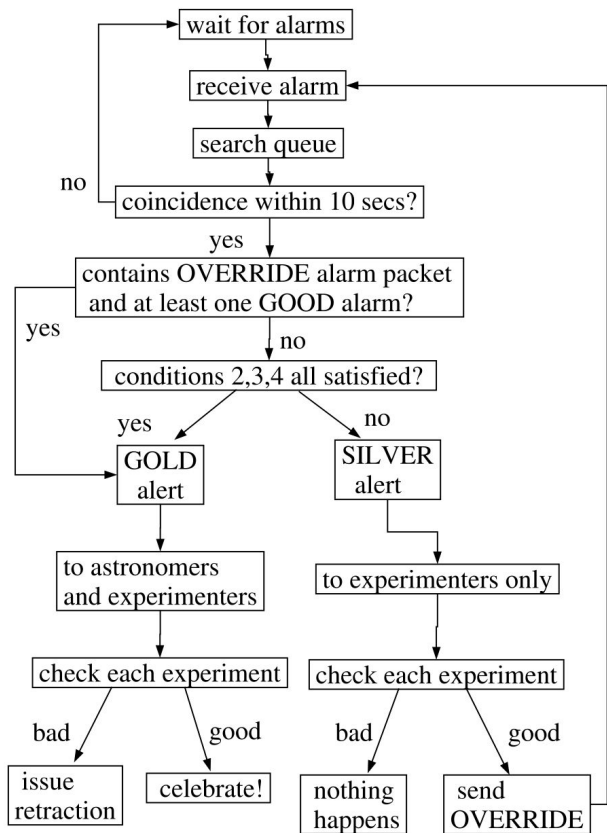
# Implementation

SNEWS can potentially use the hop tool as a backend for communication and alerts. This may make sending alerts to telescopes easier (not considered at moment). The current implementation consists of three parts:

- **deque**: a data structure based on python “collections.deque” that stores most recent messages and removes old messages.
- **decider**: an object that has **deque** as a field. Behaviors include adding messages and deciding if current messages indicate a supernova.
- **model**: interact with hopskotch and instantiate a **decider** object. Pass messages in hop.stream to the decider. Publish to TOPIC 2 if the decider decides “True”.

# Use Case with Hopskotch





# SNEWS 1.0 Alert Flowchart

**Alert:** Supernova neutrino candidate is detected by a single experiment.

→ publish to Topic 1, sent to Decider

**Alert:** Decider checks if two or more experiments pass all the coincidence requirements.

If coincidence is not met:

→ publish to Topic 2, sent back to experiments (further checking done)

If coincidence is met:

→ publish to Topic 3, sent back to experiments (further checking done) and to astronomers

# Goals & Next Steps

1. Develop prototype network with 3 experiments and one decider to test coincidence requirements.
2. Implement SNEWS 1.0 coincidence requirements to reflect what real alerts would look like.
3. Test robustness.
4. End of summer: **Make a decision to use SCiMMA & Hopskotch for SNEWS 2.0 or not**

# Demo



# Useful Links

- Hop-client: <https://github.com/scimma/hop-client>
- Hop-client demo: <https://github.com/scimma/may2020-techthon-demo/blob/master/HopkotchTutorial.md>
- Docker containers: <https://github.com/scimma/scimma-server-container>
- Apache Kafka: <https://kafka.apache.org/documentation/>
- Workshop recordings: <https://www.youtube.com/channel/UC8oQojENio8vOWT52Uy0YKA>
- SCIMMA: <https://scimma.org/>, <https://arxiv.org/pdf/1903.04590.pdf>
- SNEWS 1.0: <https://iopscience.iop.org/article/10.1088/1367-2630/6/1/114>

# SCiMMA Hopskotch messaging and data flow

