# SnapProposal

## Snap Proposal

## Abstract

*Snap* is an open telemetry framework designed to simplify the collection, processing and publishing of system data through a single API.

## Proposal

Snap provides: 1. A hardened, extensively tested daemon, snapteld, and CLI; 2. A growing number of maturing plugins; 3. Lots of example tasks to gather and publish metrics. Snap is a powerful open telemetry framework designed to:

- Improve deployment model and flexibility of the telemetry tools ecosystem
- Provide dynamic control of collection for small or large clusters of systems
- Allow flexible processing of telemetry data on agent (e.g. machine learning)
- Simplify disseminating data to telemetry ingesting systems
- Provide operational innovation for collecting across cluster of machines
- Support emerging API consumption models

## Background

In the datacenter, information is everywhere and in every form. Gathering information from disparate systems is a challenge on a single system let alone at scale. We want to achieve a signal extensible flow from raw information to valuable automation. That's why we started the work of Snap project to implement the flow. The flow includes three parts: collection, process and publish.
Collection: The workflow that starts with collection, regardless of the layer. Gather all the raw metrics of potential value we can find in our environment that may lead to informed decisions downstream. Processing: Then we add a great deal of value with the right amount of processing of that information. Perform some basic filtering or decorate our information by adding further context. Publish: This processed information is then published to one or more platforms. Our workflow should be able to treat all endpoints as the same, since they're guaranteed to change with time. This information would then be pulled into real use cases for the different parts of the business: better monitoring, orchestration and scheduling, billing/chargeback. This data is valuable in many ways to many people who can all have their own view of it. Snap is easy for integration with different systems. Our framework support pluggable connecters (plugins) to collect data from different platforms, to add more processing filter and to publish data to different systems.

## Rationale

There is a strong need for telemetry data collection in cluster environment. Snap empowers systems to expose a consistent set of telemetry data, simplify telemetry ingestion across ubiquitous storage systems, allow flexible processing of telemetry data on agent (e.g. filtering and decoration) and provide powerful clustered control of telemetry workflows across small or large clusters. Some cluster server management project (e.g. Smart Storage Management project) can use Snap project to collect required data.
Snap has integrated with other open-source projects like :

- Kubernetes(https://github.com/intelsdi-x/snap-integration-kubernetes) ;
- OpenStack (OpenStack collectors): It is used by Grafana (https://grafana.com/plugins/raintank-snap-app );
- Snap also aims at delivering data for Big Data and Machine Learning and can be used by projects like Spark or Flink;

## Current Status

### Meritocracy

Snap project was setup in 2015. The code repositories are hosted on github. We have received many contributions by reporting JIRA issues and pull requests from the community. We have clear defined rules for 'How to Contribute to the snap project" in documents (code of conduct.md and contributing. md in main repo, maintainers.md in docs folder). We have new users and committers from around the world. We encourage code commits, comments and feedbacks, and new requirements.

### Community

We have a community in Github and a group of maintainer are supporting it. We encourage new contributors in the community. By now we have 74 contributors including 28 contributors from external. We have got 1600+ stars and the fork number is ~250.

### Core Developers

The core developers are a diverse group of developers many of which are already very experienced open source developers. We have core developers from Intel and Raintank. We also have contributors as individual volunteers or from other companies like Staple, Opsview and etc. One of the initial committers, Anthony Woods, is working in Raintank. He contributed two Snap plug-ins (snap-plugin-collector-memcache and snap-plugin-collector-ping ) and integrate Snap for use in Grafana.

### Alignment

Snap is aligning with Apache project. Snap is Apache 2.0 licensed. Snap could be a good data source for big data projects like Spark. We want Snap to collect broad and diverse range of metrics, from HW to application layer - that requires engagement of the community knowledgeable in relevant areas.

## Known Risks

### Orphaned Products

The community is active. We have a group of maintainer for the project. The risk of orphaned or abandoned code is low. Snap document provide start up guide so that we believe to be able to quickly grow the developer and user communities. So Snap is less dependent on individuals. Snap has many plugin repos, and the risk is that some maybe over time and become less interesting or relevant to community and may be orphaned. This is an inevitable result of evolving and accommodating to community needs. We will figure out a plan on how to handle these over time plugins.

### Inexperience with Open Source

Snap was started as an open source project in 2015 and has remained so for 2 years. This would be non-risk for us. Founders and current maintainers are experienced in open source projects development and maintenance.

### Homogeneous Developers

The majority maintainers are from Intel. However, we have developers from several different companies (e.g. Opsview, Grafana, Mesos and etc) plus many independent volunteers. The committers are geographically distributed across the Poland, China, and US. They are experienced with working in a distributed environment.

### Reliance on Salaried Developers

Snap was started as an Intel project and the most key developers are Intel employees. However, a lot of contributors are from outside Intel. We have outside Intel 28 committers and they are from other companies like Staples, Opsview, Grafana and Raintank. We also have individual contributors to post issues and requests. The risk could be mitigated by the contributors' diversity. We look forward to more Apache developers and researchers to contribute to the project if Snap becomes an Apache project.

### Relationships with Other Apache Products

Snap is used by several Apache projects. We have metrics collectors/publishers for apache projects like: Apache http server, cassandra, kafka and mesos.

### An Excessive Fascination with the Apache Brand

We really want Snap to evolve and grow in community. Apache is the best for this. While Apache brand would definitely increase confidence and adoption of Snap in the community, our main interest is to home Snap in a mature open source community following an established development model. We would not seek to use Apache brand as a marketing tool.

## Documentation

Main information on Snap can be found at:

- https://github.com/intelsdi-x/snap/doc

Blogs and demos can be found at:

- https://medium.com/intel-sdi

## Initial Source

Snap has been under development since 2015 by a team of engineers at Intel® Corp. The initial code was centered around plugin architecture for collecting, processing and publishing metrics. It is currently hosted on Github.com under an Apache license at https://github.com/intelsdi-x/snap and other distributed repositories for plugins.

## External Dependencies

The dependencies all have Apache compatible licenses. These include Apache, BSD, and MIT licensed dependencies.

- App container (Apache License 2.0)
- Govalidator (MIT License)
- Go-yaml (Apache License 2.0)
- HttpRouter (BSD 3-clause license)
- Protobuf(BSD 3-clause license)
- Grpc (BSD 3-clause)

# Cryptography

We are not developing any cryptographic code - but using well established solutions (TLS) for data communication protection (for management functions - REST API, and data exchange - TLS for plugin communication).

# Required Resources

## Mailing Lists

- private@snap.incubator.apache.org
- dev@snap.incubator.apache.org (snap-dev)
- commits@snap.incubator.apache.org (snap-commits)
- user@snap.incubator.apache.org (snap-user)

## Subversion Directory

Git is the preferred source control system: git://git.apache.org/Snap

## Git repository

- [https://github.com/apache.org/snap](https://github.com/apache.org/snap)

## Issue Tracking

JIRA Snap (SNAP)

# Initial Committers

- Andrzej Kuriata <andrzej.kuriata at intel dot com>
- Izabella Raulin <izabella.raulin at Intel dot com>
- Katarzyna Kujawa <katarzyna.kujawa at Intel dot com>
- Fengfeng Tao <fengfeng.tao at intel dot com>
- Jialei Wang <jialei.a.wang at intel dot com>
- Yajuan Shi <yajuan.shi at Intel dot com>
- Wenjun Zeng <wenjun.zeng at Intel dot com>
- Dong Zhao <dong.zhao at Intel dot com>
- Dancy Ding <dancy.ding at Intel dot com>
- Anthony Woods <awoods at grafana dot com>

# Affiliations

- Andrzej Kuriata <intel>
- Izabella Raulin <Intel>
- Katarzyna Kujawa <Intel>
- Fengfeng Tao <Intel>
- Jialei Wang <Intel>
- Yajuan Shi <Intel>
- Wenjun Zeng <Intel>
- Dong Zhao <Intel>
- Dancy Ding <Intel>
- Anthony Woods <Grafana, Raintank>

# Sponsors

## Champion

TBD

## Nominated Mentors

- Luke Han (Apache Member)
- Uma Gangumalla (Apache Member)
- Andrew Purtell (Apache Member)

## Sponsoring Entity

We are requesting the Incubator to sponsor this project.