

# QuarksProposal

## Quarks Proposal

### Abstract

Quarks is a stream processing programming model and lightweight runtime to execute analytics at devices on the edge or at the gateway.

### Proposal

- Quarks is a programming model and runtime for streaming analytics at the edge. Applications are developed using a functional flow api to define operations on data streams that is executed as a graph of "oplets" in a lightweight embeddable runtime. The SDK provides capabilities like windowing, aggregation and connectors with an extensible model for the community to expand its capabilities.

### Background

- Stream processing systems are commonly used to process data from edge devices and there is a need to push some of the streaming analytics to the edge to reduce communication costs, react locally and offload processing from the central systems. Quarks was developed by IBM as an entirely new project to provide an SDK and lightweight embeddable runtime for streaming analytics at the edge. Quarks was created to be an open source project that could provide edge analytics to a broad community and foster collaboration on common analytics and connectors across a broad ecosystem of devices.

### Rationale

- With the growth in number of connected devices (Internet of Things) there is a need to execute analytics at the edge in order to take local actions based upon sensor information and/or reduce the volume of data sent to back-end analytic systems to reduce communication cost. Quarks rationale is to provide consistent and easy to use programming models to allow application developers to focus on their application rather than issues like device connectivity, threading etc. Quarks' functional data flow programming model is similar to systems like Apache Flink, Beam (An incubating Apache project), Java 8 Streams & Apache Spark. The API currently has language bindings for Java8, Java7 and Android. Quarks was developed to address requirements for analytics at the edge for IoT use cases that were not addressed by central analytic solutions. We believe that these capabilities will be useful to many organizations and that the diverse nature of edge devices and use cases is best addressed by an open community. Therefore, we would like to contribute Quarks to the ASF as an open source project and begin developing a community of developers and users within Apache.

### Initial Goals

- Quarks initial code contribution provides:
- APIs for developing applications that execute analytics using a per-event (data item) streaming paradigm including support for windows against a stream for aggregation
- A micro-kernel style runtime for execution.
- Connectors for MQTT, HTTP, JDBC, File, Apache Kafka & IBM Watson IoT Platform
- Simple analytics aimed at device sensors (using Apache Common Math)
- Development mode including a web-console to view the graph of running applications
- Testing mechanism for Quarks applications that integrates with assertion based testing systems like JUnit
- Android specific functionality such as producing a stream that contains a phone's sensor events (e.g. ambient temperature, pressure)
- JUnit tests .
- All of the initial code is implemented using Java 8 and when built produces jars that can execute on Java 8, Java 7 and Android. The goal is to encourage community contributions in any area of Quarks, to expand the community (including new committers) and use of Quarks. We expect contributions will be driven by real-world use of Quarks by anyone active in the IoT space such as auto manufactures, insurance companies, etc. as well as individuals experimenting with devices such as Raspberry Pis, Arduinos and/or smart phone apps etc. Contributions would be welcomed in any aspect of Quarks including: .
- Support for additional programming languages used in devices such as C, [OpenSwift](#), Python etc.
- Specific device feature (e.g. Raspberry Pi, Android) or protocol (e.g. OBD-2) support
- Connectors for device to device (e.g. [AllJoyn](#)), device local data sources, or to back-end systems (e.g. a IoT cloud service)
- Additional analytics, either exposing more functionality from Apache Common Math, other libraries or hand-coded analytics.
- Improvements to the development console, e.g. additional visualizations of running applications
- Documentation, improving existing documentation or adding new guides etc.
- Sample applications
- Testing
- . The code base has been designed to be modular so that additional functionality can be added without having to learn it completely, thus new contributors can get involved quickly by initially working on a focused item such as an additional analytic or connector. The only constraints on contributions will be to keep Quarks on its focus of IoT and edge computing, with attributes such as small footprint and modularity to allow deployments to only include what is needed for that specific device and/or application.

### Current Status

- Quarks is a recently released project on Github <http://quarks-edge.github.io>. The current code is alpha level code but is functional and has some basic tests. The team is looking forward to working in the Apache community to enhance the functionality to allow robust streaming of devices on the edge.

### Meritocracy

- Quarks was originally created by Dan Debrunner, William Marshall, Victor Dogaru, Dale [LaBossiere](#) and Susan Cline. We plan to embrace meritocracy and encourage developers to participate and reach committer status. Dan Debrunner was the initial creator of the Apache Derby code and a committer when Derby was accepted into incubation. He is an Apache member and has experience with the Apache Way. Derby is a successful project that embraces the Apache meritocracy and graduated from incubation with a diverse group of committers. .
- With an abundance of devices that potentially can take advantage of Quarks, there is a large pool of potential contributors and committers. The initial team is enthusiastic about assisting and encouraging involvement.

## Community

- Quarks currently has a very small community as it is new, but our goal is to build a diverse community at Apache. The team strongly believes that a diverse and vibrant community is critical as devices on the edge vary quite a bit. The community will benefit from developers who have expertise in various devices. We will seek to build a strong developer and user community around Quarks.

## Core Developers

- The initial developers have many years of development experience in stream processing. The initial development team includes developers who have experience with Apache, including one Apache member, and with other open source projects on Github.

## Alignment

- Quarks interacts with other Apache solutions such as Apache Kafka and Apache Spark. Quarks is API driven, modular and written in 100% java, making it easy for developers to pick up and get involved.

## Known Risks

### Orphaned products

- The contributors are from a leading vendor in this space, who has shown a commitment to Apache projects in the past. They are committed to working on the project at least for the next several years, as the community grows and becomes more diverse.

### Inexperience with Open Source

- Several of the core developers have experience with Apache, including a developer who is a committer on Derby and an Apache member. All of the core developers have some level of experience with the use of open source packages and with contributions on projects on sites such as [Github](#).

### Homogenous Developers

- The initial set of developers come from one company, but we are committed to finding a diverse set of committers and contributors. The current developers are already very familiar with working with many geographies, including developers in most geographies around the world. They are also very comfortable working in a distributed environment.

### Reliance on Salaried Developers

- Quarks currently relies on salaried developers at this time, but we expect that Quarks will attract a diverse mix of contributors going forward. For Quarks to fully transition to an "Apache Way" governance model, we will embrace the meritocracy-centric way of growing the community of contributors.

### Relationships with Other Apache Products

- These Apache projects are used by the current codebase: .
- Apache Ant - Build
- Apache Common Math - Initial analytics
- Apache HTTP Components [HttpClient](#) - HTTP connectivity
- Apache Kafka - Kafka is supported as a message hub between edge Quarks applications and back-end analytics systems Events from Quarks applications sent through message hubs (such as Apache Kafka) may be consumed by back-end systems such as Apache Flink, Apache Spark, Apache Samza, Apache Storm, Beam (in incubation) or others.

### A Excessive Fascination with the Apache Brand

- Quarks will benefit greatly from wide collaboration with developers working in the device space. We feel the Apache brand will help attract those developers who really want to contribute to this space. Several developers involved with this project have a very positive history with Derby and feel that Apache is the right place to grow the Quarks community. We will respect Apache brand policies and follow the Apache way.

## Documentation

- <http://quarks-edge.github.io/quarks.documentation>

## Initial Source

- Quarks code has been recently released on Github under the Apache 2.0 license at <https://github.com/quarks-edge/quarks> . It was created by a small team of developers, and is written in Java.

## Source and Intellectual Property Submission Plan

- After acceptance into the incubator, IBM will execute a Software Grant Agreement and the source code will be transitioned to the Apache infrastructure. The code is already licensed under the Apache Software License, version 2.0. We do not know of any legal issues that would inhibit the transfer to the ASF.

## External Dependencies

- The dependencies all have Apache compatible license. These include Apache, MIT and EPL. The current dependencies are:
- D3
- Jetty
- Apache Kafka
- Metrics
- MQTTV3
- SLF4J
- GSON
- Apache commons Math3 .
- Development tools are .
- Java SDK 8
- Eclipse 4.5
- Ant 1.9
- Junit 4.10

## Cryptography

- No cryptographic code is involved with Quarks.

## Required Resources

### Mailing lists

- [private@quarks.incubator.apache.org](mailto:private@quarks.incubator.apache.org) (with moderated subscriptions)
- [dev@quarks.incubator.apache.org](mailto:dev@quarks.incubator.apache.org)
- [commits@quarks.incubator.apache.org](mailto:commits@quarks.incubator.apache.org)

### Git Repository

- <https://git-wip-us.apache.org/repos/asf/incubator-quarks.git>

### Issue Tracking

- Jira Project Quarks (QUARKS)

### Other Resources

- Means of setting up regular builds and test cycle.

## Initial Committers

- Daniel Debrunner: [djd at apache dot org](mailto:djd@apache.org) - CLA on file
- Susan Cline: [home4slc at pacbell dot net](mailto:home4slc@pacbell.net) - CLA on file
- William Marshall: [wcmarsha at gmail dot com](mailto:wcmarsha@gmail.com) - CLA on file
- Victor Dogaru: [vdogaru at gmail dot com](mailto:vdogaru@gmail.com) - CLA on file
- Dale [LaBossiere](#): [dml.apache at gmail dot com](mailto:dml.apache@gmail.com) - CLA on file

## Affiliations

- Daniel Debrunner IBM
- Susan Cline IBM
- William Marshall IBM
- Victor Dogaru IBM
- Dale Labossiere IBM

## Additional Interested Contributors

- May Wone: [mnwone at gmail dot com](mailto:mnwone@gmail.com)
- Sandeep Deshmukh: [sandeep at datatorrent dot com](mailto:sandeep@datatorrent.com)
- Bhupesh Chawda: [bhupeshchawda at gmail dot com](mailto:bhupeshchawda@gmail.com)

## Sponsors

**Champion**

- Katherine Marsden (kmarsden at apache dot org)

**Nominated Mentors**

- Katherine Marsden (kmarsden at apache dot org)
- Daniel Debrunner (djd at apache dot org)
- Luciano Resende (lresende at apache dot org)
- Justin Mclean (justin at classsoftware dot com)

**Sponsoring Entity**

- The Incubator