# Notorious F.A.S.T.

High-Speed High-Volume Big Data Plumbing

Tribute to Notorious B.I.G.

# Features

- High-speed data reception
- Storage and notification techniques that enable real-time display of data

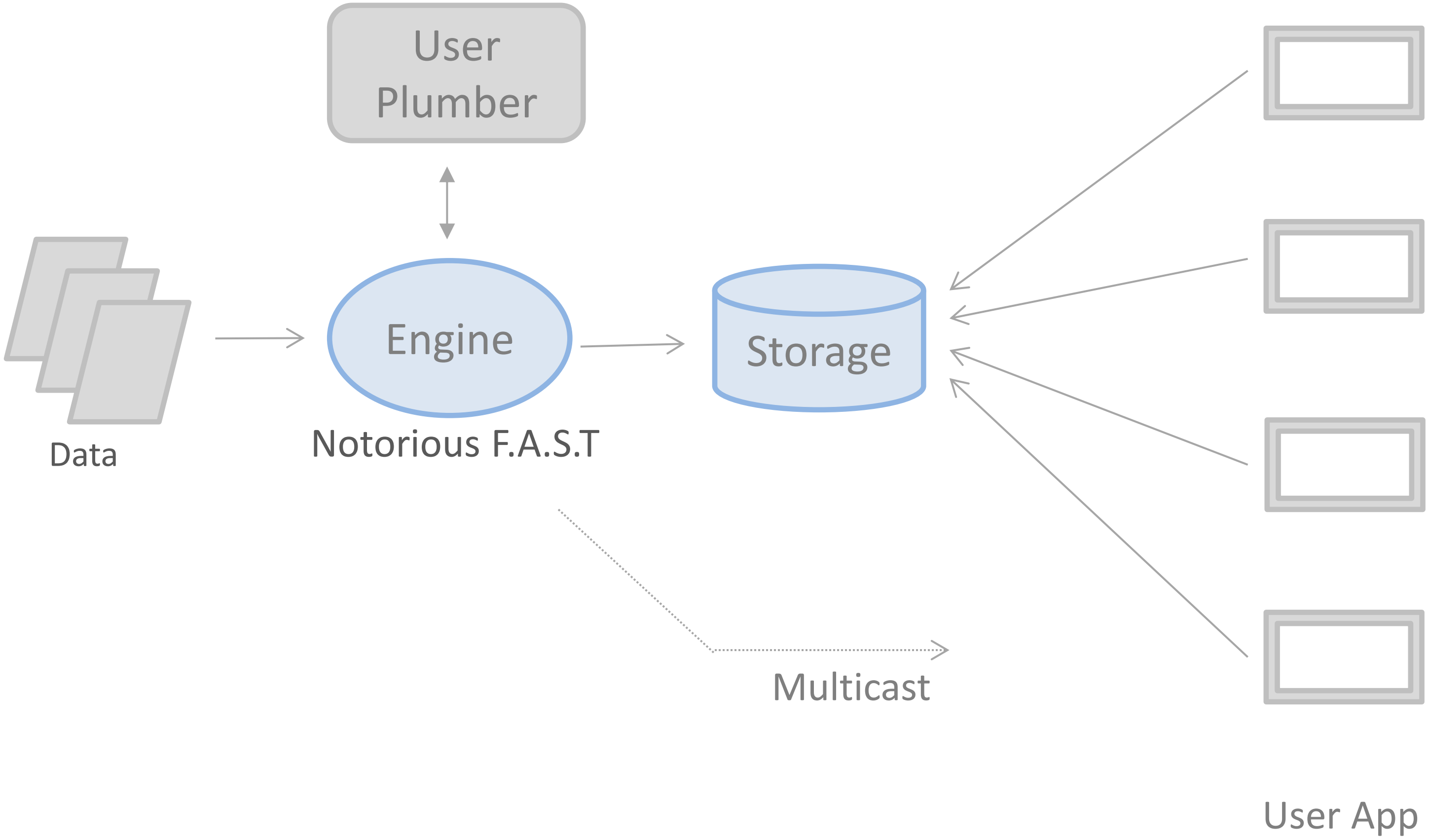
# Goals

- Designed as simple as possible
- Concurrent as much as possible
- Genuine stability due to simple design
- Flexibility in application building

# Applications

- Weather data
- Traffic data
- Satellite data
- SNS postings
- Locations of mobile phones
- Stock and currency price data
- Air traffic control data

# Configuration



# How it Works

- A queue is setup per data source.
- A queue handler thread reads data from a queue and spawns a worker thread.
- A piece of data can contain high priority and lower priority items.
- High priority items are immediately sent to display applications through UDP multicast.
- Lower priority items are saved to a data store for pickup by app when needed.

# User Components

- User data
- User-written plumber that modifies data with a help of plumbing engine
- User-written display apps

  – Language supported: Java, C#, Qt/C++, Python, …

# Provided Components

- Notorious F.A.S.T.
  High-speed multi-threading plumbing engine
  - Receive, organize, notifies and save data …
  - in a fashion suitable for quick display by app
- Multiple selection of data storage
  - Hadoop, SQL, Redis and others
- Display app support library
  - for quick data retrieval by display app

# Technologies

- Blocking queue
- Multicast
- Concurrency
  - java.util.concurrent
  - QtConcurrent
  - Qt event loop on a different thread
  - .NET Asynchronous
  - …

- Introduction to Notorious F.A.S.T.
- 2016/06/04 Written
- 2016/06/19 Updated

- Java concept code available at github.com/jjking2/notfast