IoT Data Connector - Fluent Bit

#iotlt

Toru Takahashi

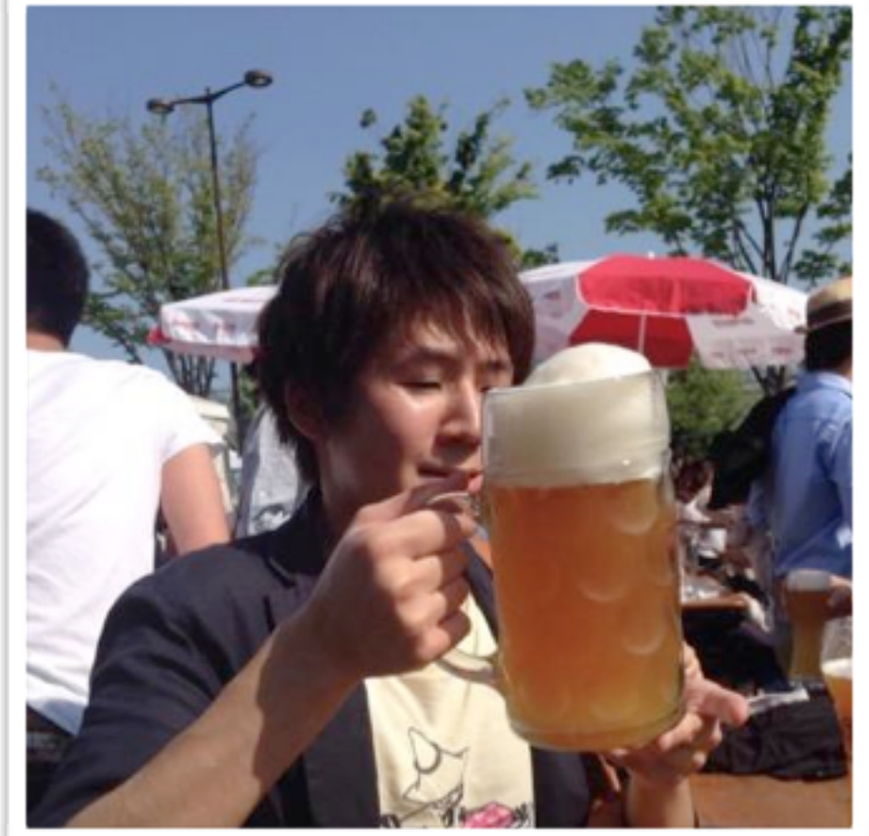
June 17, 2015

IoTLT vol.4

TREASURE

About Me

- **Toru Takahashi**
  - twitter: @nora96o
  - github: toru-takahashi

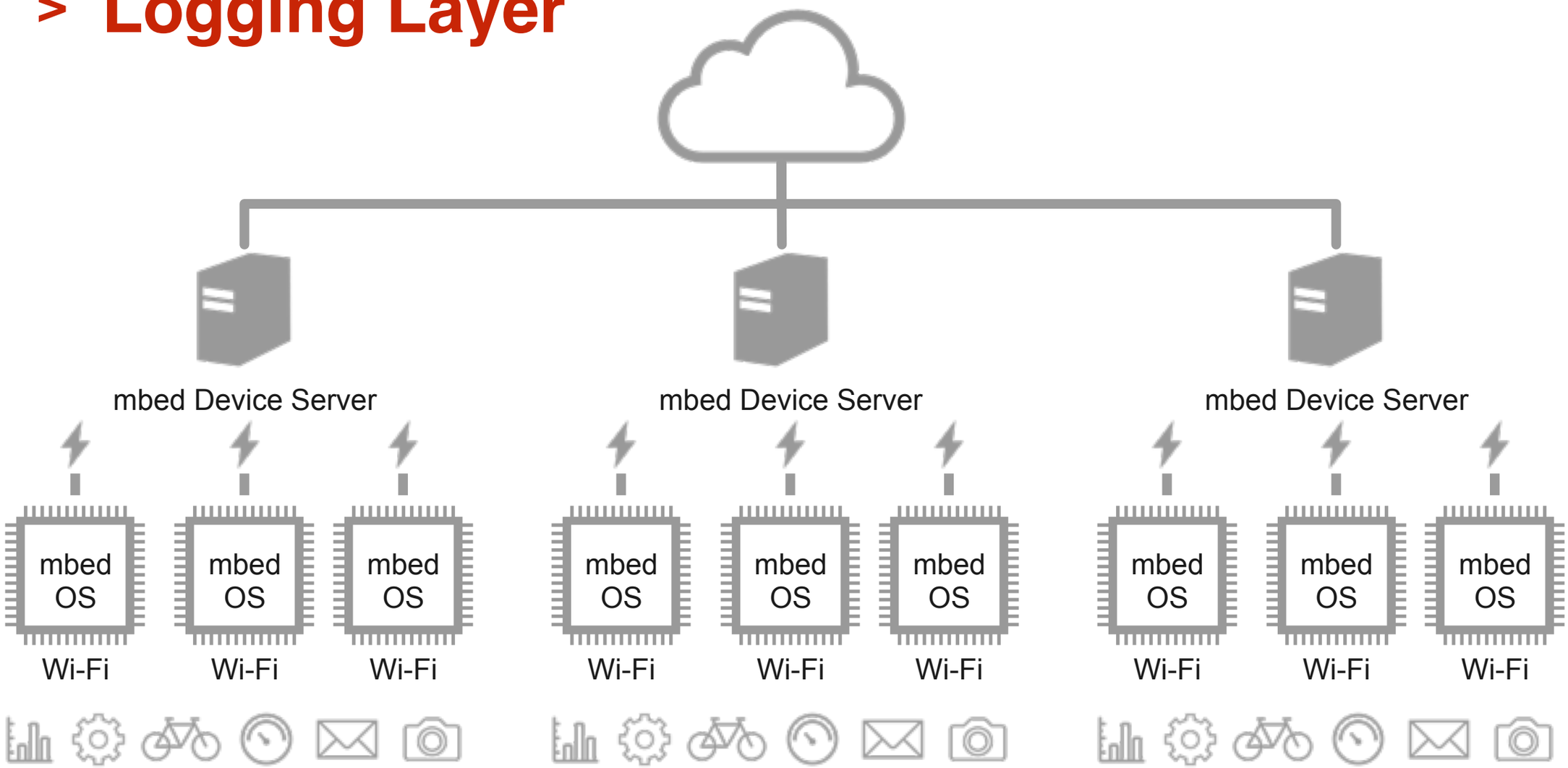


- **Treasure Data, Inc.**
  - Technical Support Engineer

- **I love working in Maid café :)**
  - **Most of staying in 女中酒場幻橙館**

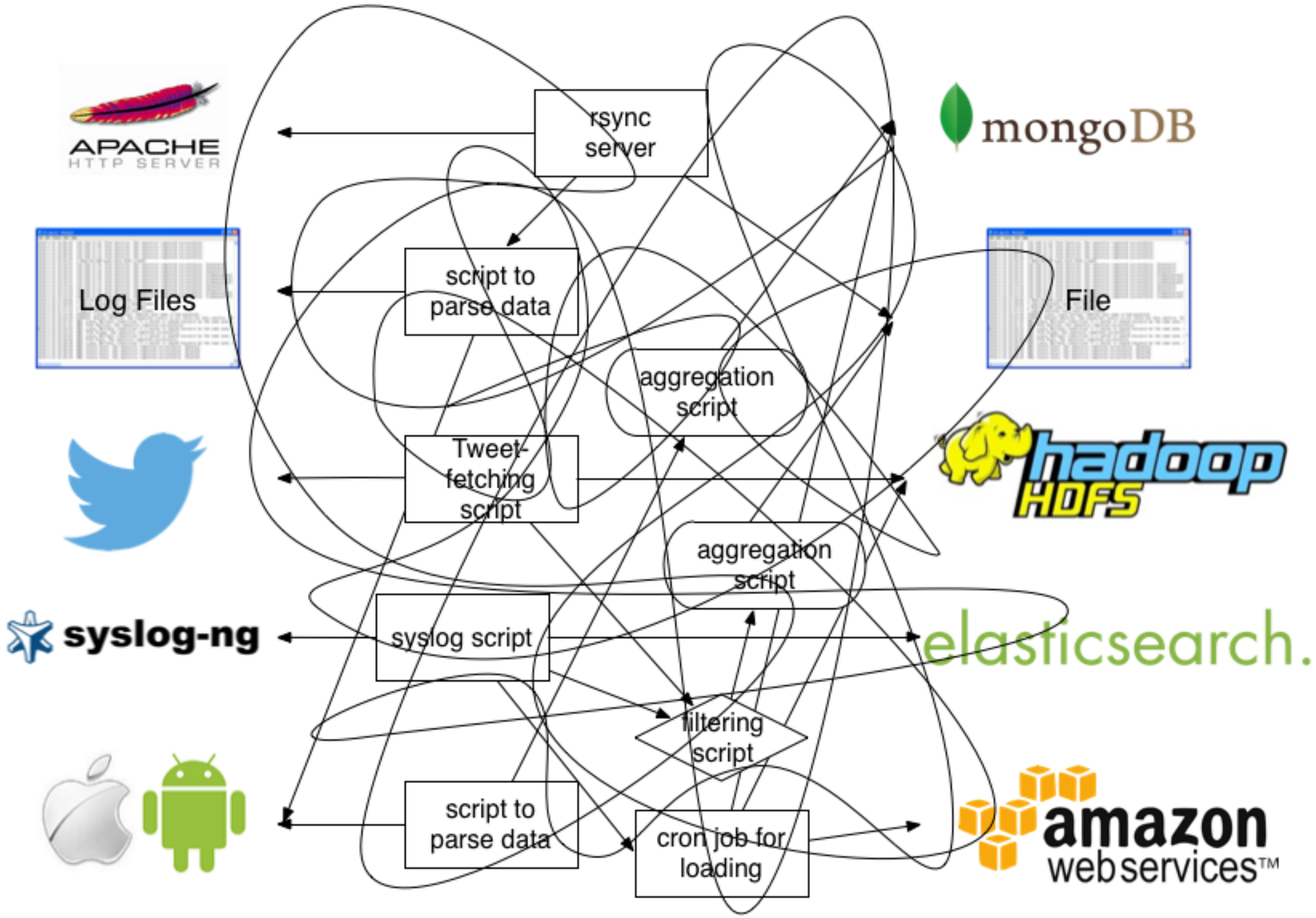
Today's Talk focus … on IoT

- ~~Application Layer~~
- Logging Layer



Logging Layer on Web

Before


APACHE
HTTP SERVER
rsync server
mongoDB
Log Files
script to parse data
File
aggregation script
Tweet-fetching script
hadoop HDFS
aggregation script
syslog-ng
syslog script
elasticsearch.
filtering script
script to parse data
cron job for loading
amazon web services™

fluentd

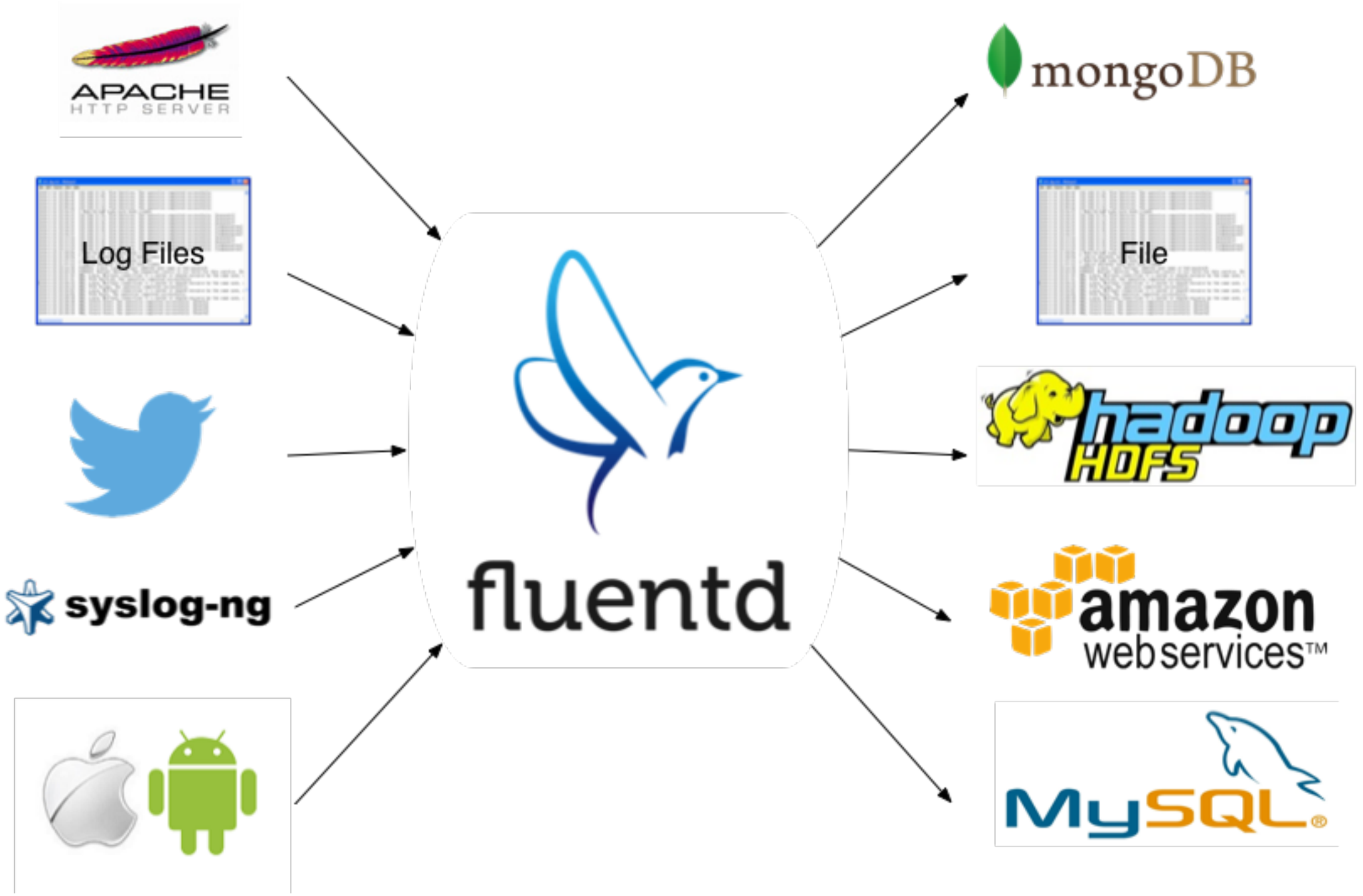
Structured logging

Reliable forwarding

Pluggable architecture

http://fluentd.org/

After

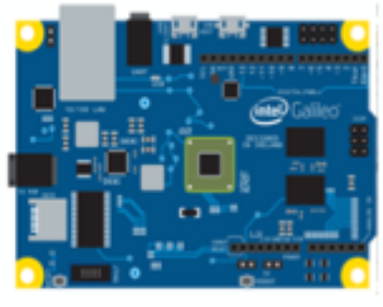

APACHE
HTTP SERVER
Log Files
syslog-ng
fluentd
mongoDB
File
hadoop
HDFS
amazon
web services™
MySQL

What's Fluentd?

- **Data collector for unified logging layer**
  - Streaming data transfer based on **JSON**
  - Written in **Ruby**
- **Gem based various plugins**
  - http://www.fluentd.org/plugins
- **Working in production**
  - http://www.fluentd.org/testimonials

Logging Layer on IoT

Current



IoTivity

AllJoyn™

mongoDB

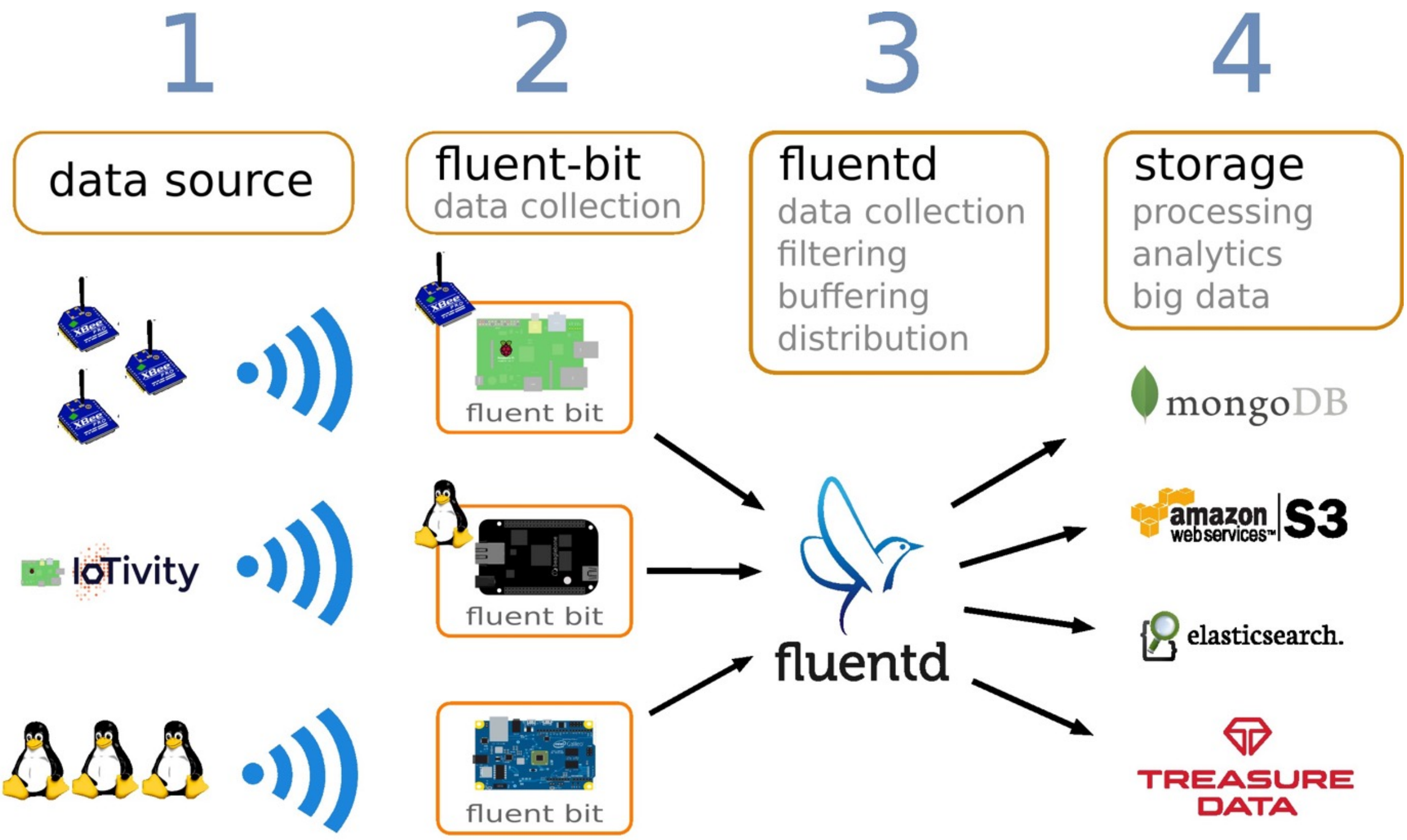
amazon web services™ | S3

elasticsearch.

TREASURE DATA

Fluent Bit

Future


1
2
3
4
data source
fluent-bit
data collection
fluentd
data collection
filtering
buffering
distribution
storage
processing
analytics
big data
fluent bit
fluent bit
fluent bit
IoTivity
fluentd
mongoDB
amazon web services S3
elasticsearch.
TREASURE DATA

Fluent Bit

Open Source data collection tool for Embedded Linux

- > **Sensors**
- > **Services**
- > **Signals / Radios**
- > **Operating System Info**
- > **Automotive / Telematics**

Committer: Eduardo Silva (@edsiper)

- https://github.com/fluent/fluent-bit
- http://fluentbit.io (official web site)

Fluent Bit

Features

- Collection & Distribution
  - Support Custom Input / Output
    - Configurable (through file system files)
- Built-in system metrics
- C API for Developers (WIP)
- Integration with third party services
- Open Source / Apache License v2.0

Examples

Built-in Metrics:
CPU Usage

```
$ ./fluent-bit -i cpu -o stdout
[2015/06/14 12:13:35] [ info] Flush buf 120 bytes
[0] {"time"=>1434284015, "cpu"=>12.000000}
[1] {"time"=>1434284016, "cpu"=>100.000000}
[2] {"time"=>1434284017, "cpu"=>100.000000}
[3] {"time"=>1434284018, "cpu"=>100.000000}
[4] {"time"=>1434284019, "cpu"=>82.000000}
[4] {"time"=>1434284024, "cpu"=>0.000000}
```

Built-in Metrics:
Kernel Log Message

```
$ ./fluent-bit -i kmsg -o stdout
[2015/06/14 12:19:00] [ info] starting engine
[0] {"time"=>1434283478, "priority"=>6, "sequence"=>0, "sec"=>0, "usec"=>0,
"msg"=>"Initializing cgroup subsys cpuset"}
[1] {"time"=>1434283478, "priority"=>6, "sequence"=>1, "sec"=>0, "usec"=>0,
"msg"=>"Initializing cgroup subsys cpu"}
[2] {"time"=>1434283478, "priority"=>6, "sequence"=>2, "sec"=>0, "usec"=>0,
"msg"=>"Initializing cgroup subsys cpuacct"}
[3] {"time"=>1434283478, "priority"=>5, "sequence"=>3, "sec"=>0, "usec"=>0,
"msg"=>"Linux version 3.14.35-28.38.amzn1.x86_64 (mockbuild@gobi-
build-64012) (gcc version 4.8.2 20140120 (Red Hat 4.8.2-16) (GCC) ) #1 SMP
Wed Mar 11 22:50:37 UTC 2015"}
```

Output to TreasureData:
Config File

```
[TD]
    # API
    # ===
    # The TreasureData API key. To obtain this please log into your
    API      SOME_API_KEY

    # Database
    # ========
    # Specify the name of your database, it must exists.
    Database db_example

    # Table
    # =====
    # Specify the database table name where the records will be stored
    Table    table_example
```

Roadmap

- Library mode
- Support a stock of sensors (inputs)
- HTTP input
- Release first stable version
- Documentation

IoT

https://github.com/fluent/fluent-bit

TREASURE