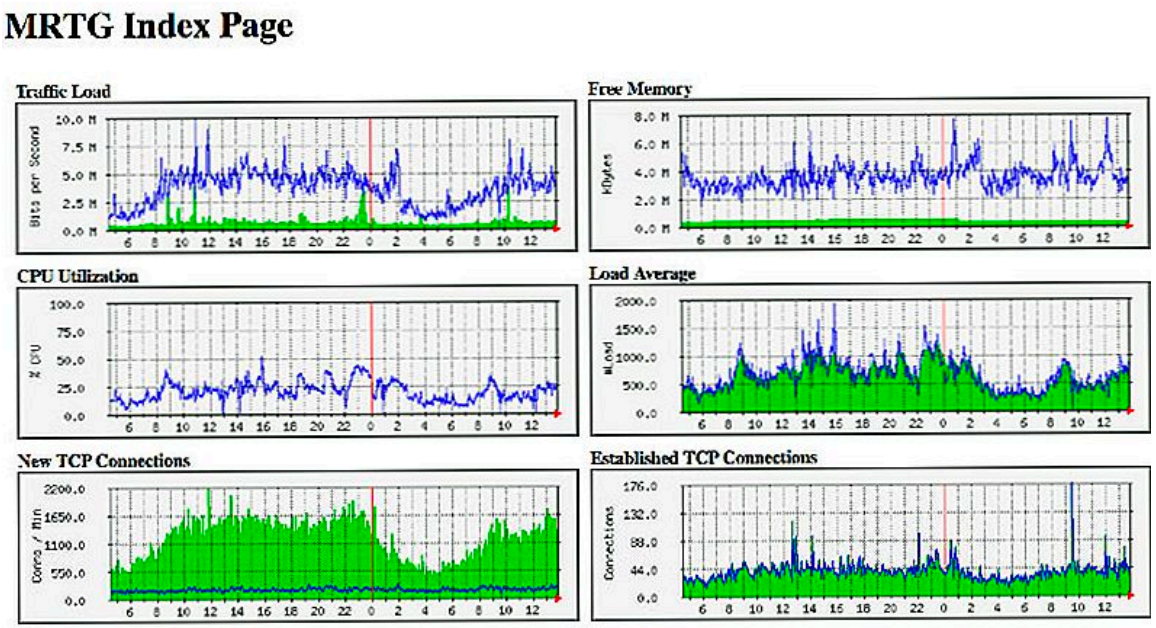
LOGICALIS

Business and technology working as one

Telegraf, InfluxDB, Grafana Training

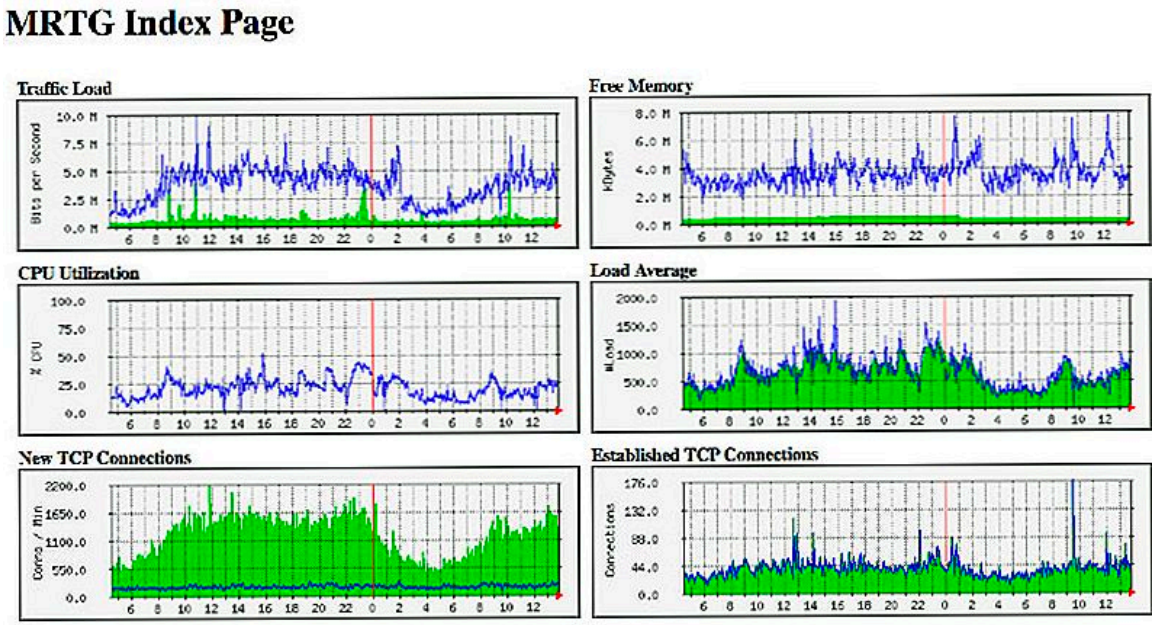
Still Using MRTG?

- Simple all in one SNMP monitoring software
  - Send SNMP requests
  - Store replies into text-based database
  - Generate images and HTML pages
- Measures two values (input / output)
- Collects data every five minutes
- Static pages
- RRDTools, Cacti



Why should not use MRTG anymore

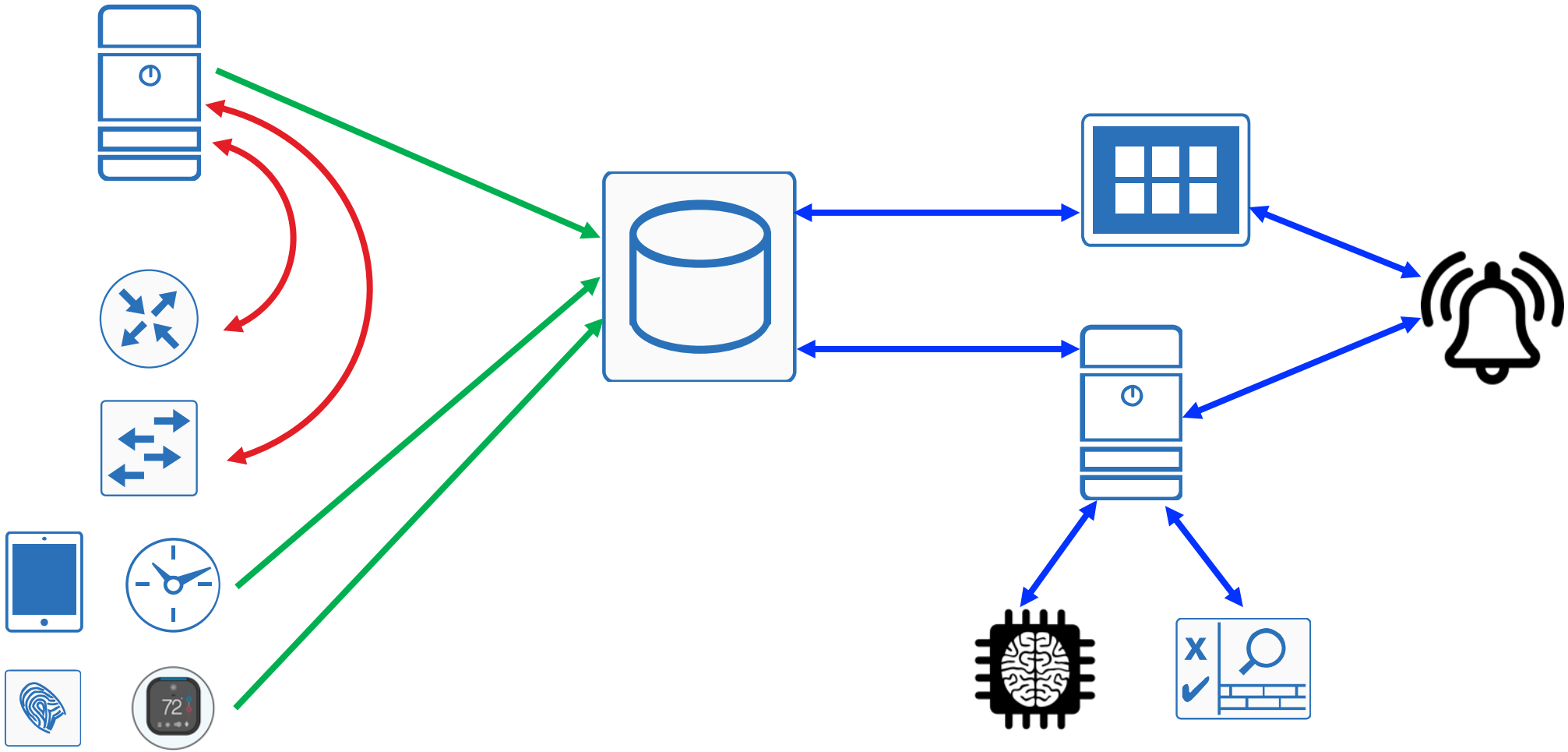
- Pull-based
- Mainly SNMP, 2-D data
- Not scalable
- Static image, web page
- Five minutes interval
- Difficult to customize
- No modern alert mechanism
- No distributed databases



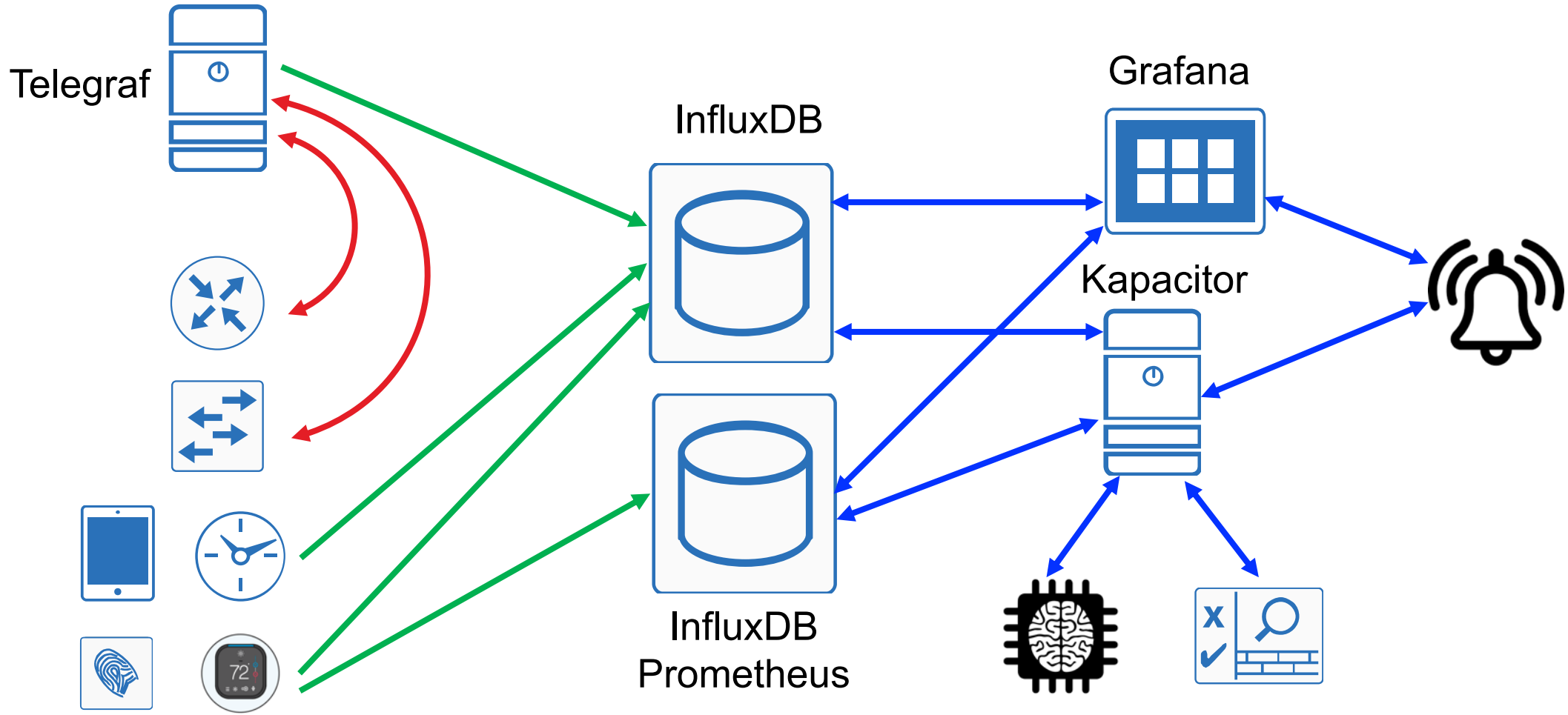
What we need

- Collect data
- Store and process data
- Visualize data
- Monitoring and alert
- Telemetry data more than SNMP
  - What is telemetry data?
  - Getting more important
  - Big Data to AI

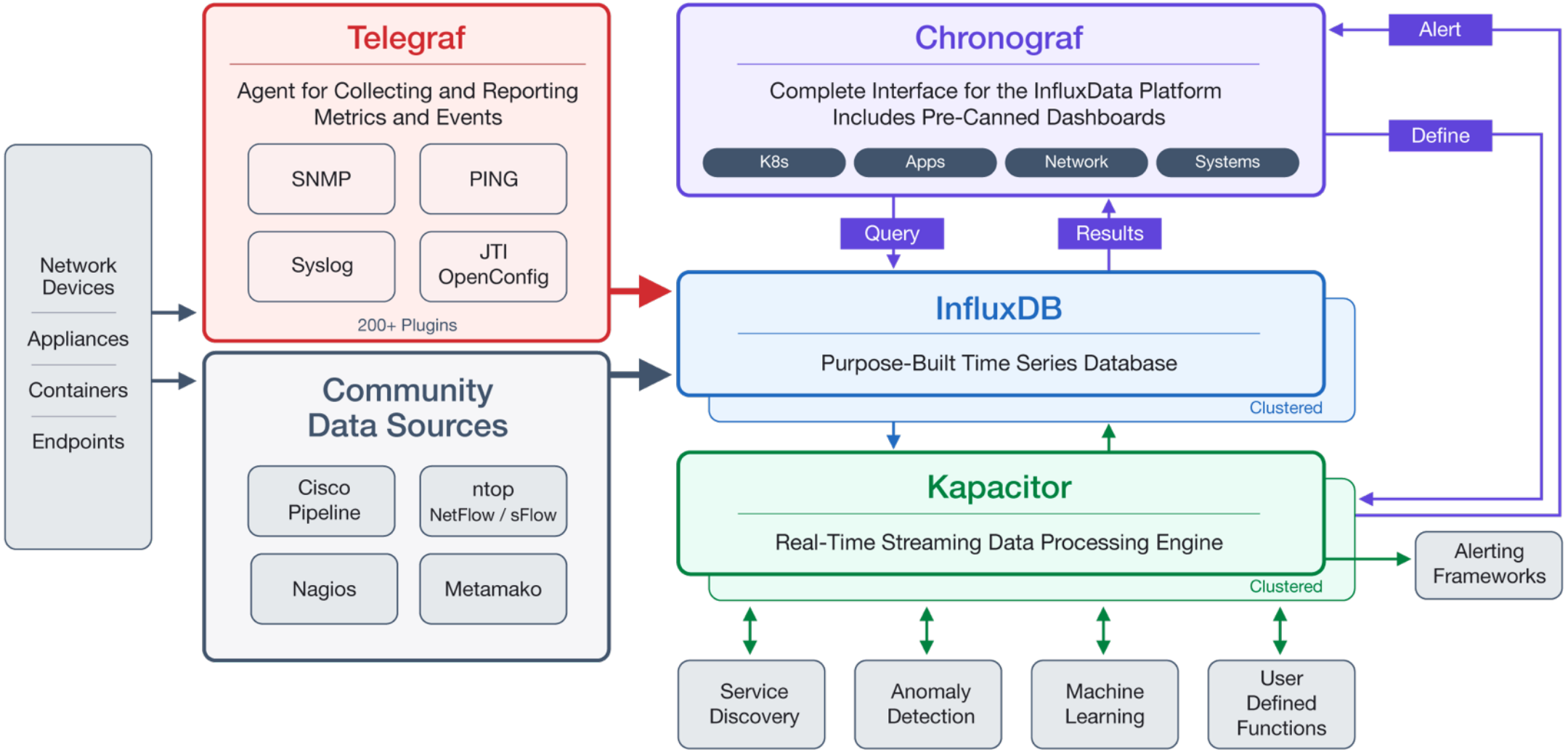
Modern Data Monitoring and Processing Model



Modern Data Monitoring and Processing Model



TICK Architecture



LOGICALIS

Products

Telegraf	InfluxDB	Chronograf	Kapacitor
Agents for collecting and reporting metrics and events	Time Series Database	Data visualization	Streaming data processing enging
Logstash Prometheus Fluentd	Graphite Prometheus OpenTSDB Elasticsearch	Grafana Kibana Datadog Splunk	Kafka Grafana Prometheus

Why InfluxDB?

Rank Nov 2019	Rank Oct 2019	Rank Nov 2018	DBMS	Database Model	Score Nov 2019	Score Oct 2019	Score Nov 2018
1.	1.	1.	IninfluxDB	Time Series	19.93	+0.31	+6.29
2.	2.	2.	Kdb+	Time Series, Multi-model	5.29	-0.15	+0.44
3.	3.	↑ 6.	Prometheus	Time Series	3.64	+0.04	+1.69
4.	4.	↓ 3.	Graphite	Time Series	3.32	-0.02	+0.48
5.	5.	↓ 4.	RRDtool	Time Series	2.90	+0.19	+0.18
6.	6.	↓ 5.	OpenTSDB	Time Series	2.13	+0.21	+0.11
7.	7.	7.	Druid	Multi-model	1.79	-0.05	+0.43
8.	8.	8.	TimescaleDB	Time Series, Multi-model	1.73	+0.22	+1.19
9.	↑ 11.	↑ 13.	FaunaDB	Multi-model	0.61	+0.14	+0.40
10.	10.	↑ 14.	GridDB	Time Series, Multi-model	0.57	+0.03	+0.40

source: https://db-engines.com/en/ranking/time+series+dbms

Why InfluxDB, Telegraf, Grafana

InfluxDB

- High performance, written in Go
- Native HTTP API
- Powerful SQL-like language
- Supports logs
- Down sampling

Telgraf

- High performance, written in Go
- Collect and send almost all kinds of data
- 200+ input, output plugins

Grafana

- Rich data sources support
  - InfluxDB, Prometheus, MySQL
- Templating
- Alerts
- Plugin, App

Time Series Data

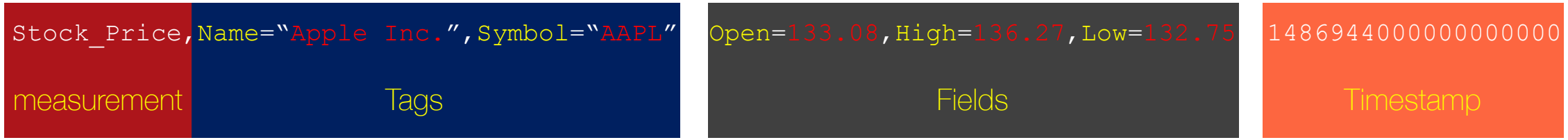
Measurement	Stock_Price
Tag	Name=Apple Inc. Symbol=AAPL
Fields	Open=133.08 High=136.27 Low=132.75 Close=135.72 Volume=136.67M Change=14.12%
Timestamp	2/13/2017

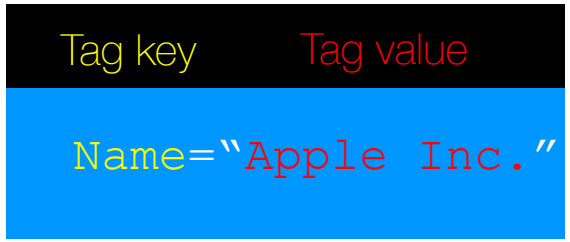
LOGICALIS

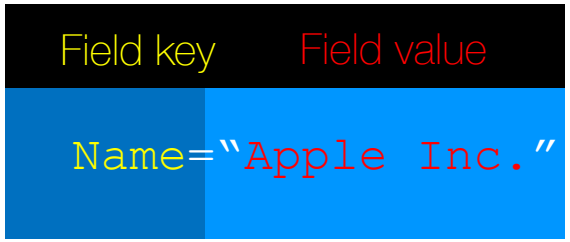
InfluxDB Data Format

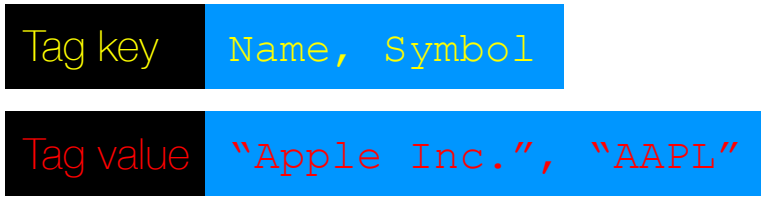
measurement	Tags	Fields	Timestamp
Stock_Price	Name="Apple Inc.",Symbol="AAPL"	Open=133.08,High=136.27,Low=132.75	1486944000000000000
CiscoSwitch	ifIndex=1,ifAlias="Gi0/1"	ifInOctets=133,ifOutOctes=136,ifStatus=1	1487244000000000000
HPE_Servers	dc="TW01",sensor="sysCpu"	user=13,system=26,idle=55,kernel=5,irq=1	1487434000000000000

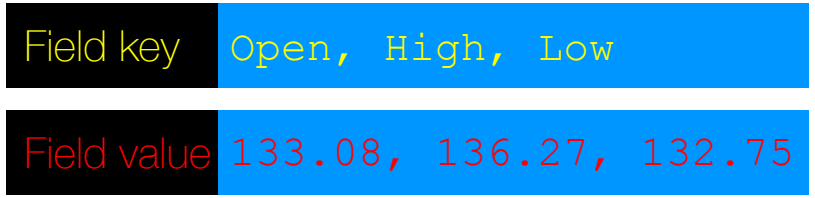
Key-Value Pairs











LOGICALIS

InfluxDB and Telegraf Configuration

InfluxDB

- Default configuration directory
- /etc/influxdb
- Default binding port: 8086
- Enable authentication (recommended)
- https://github.com/influxdata/influxdb

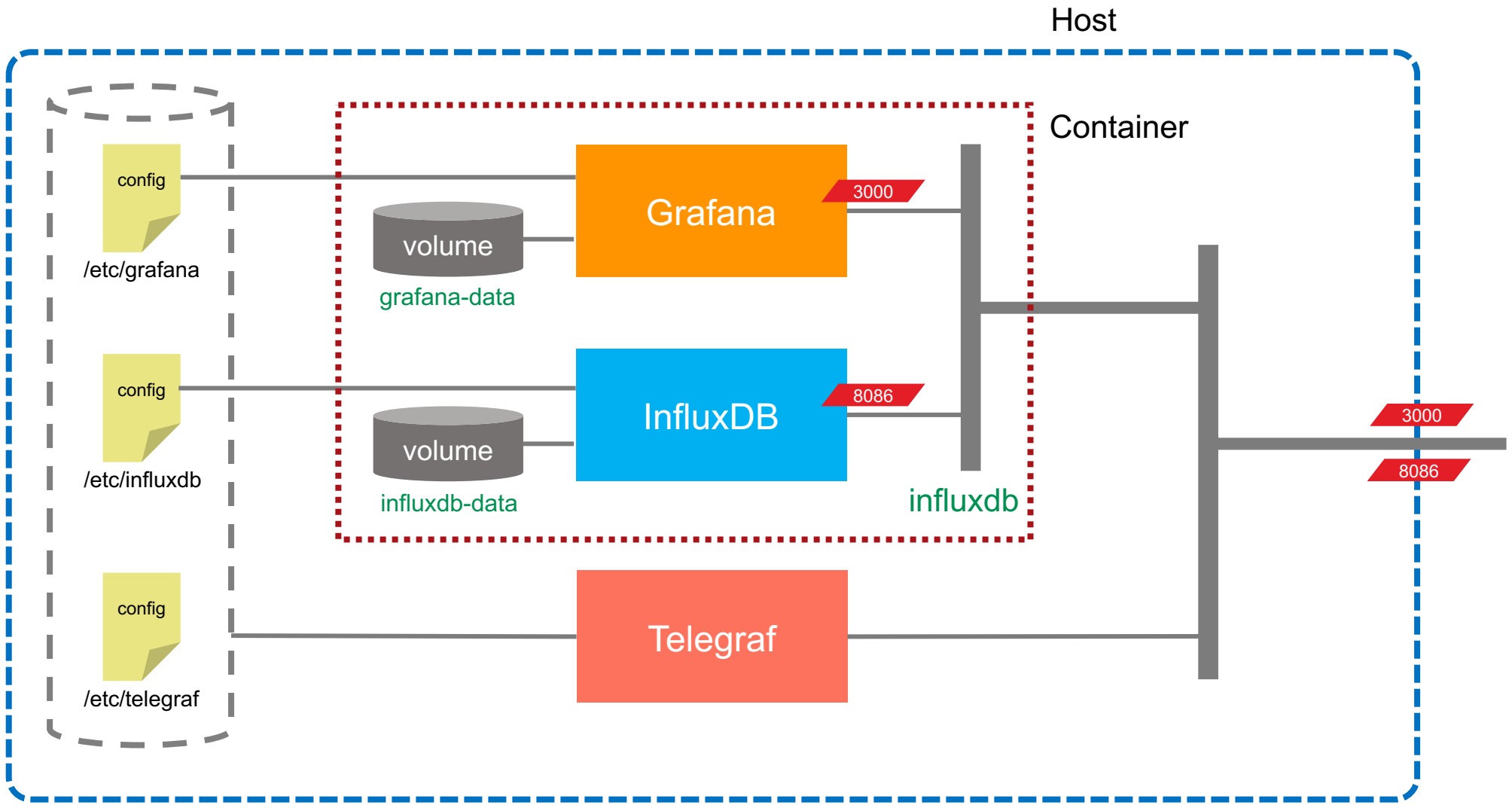
Telegraf

- Default configuration directory
  - /etc/telegraf
  - /etc/telegraf/telegraf.d
- Telegraf will load every file in the direcotry
- First, configure global parameters
  - interval, debug, logfile
- Then configure input and output plugins
- https://github.com/influxdata/telegraf

Grafana Features

- Data source
- Dashboard
- Panel
- Metrics
- Query
- Plugin
- Template
- Variable
- User
- Playlist
- Alert

Lab


Host
Container
config
/etc/grafana
volume
grafana-data
Grafana
3000
config
/etc/influxdb
volume
influxdb-data
InfluxDB
8086
influxdb
config
/etc/telegraf
Telegraf
3000
8086

Docker Commands in This Lab

Command	Comment
`docker run -dit -v --restart --rm -p --net --name`	-d: detach -it: interactive terminal -v: mount storage --restart: restart policy --rm: delete container after exit --p: publish ports --net: use network --name: name of the container
`docker exec -it`	Execute command in container
`docker network create`	Create a container network
`docker volume create`	Create a container volume

Docker Commands in This Lab

Command	Comment
`docker container cp`	Copy files in container to host
`docker container ls`	List files in container
`docker container [start | stop | restart]`	Start/Stop/Restart a container
`docker images`	List docker images
`docker rmi`	Delete docker image

InfluxDB Commands in This Lab

Command	Comment
`influxd config`	Display configuration file
`influx -username -password`	Enter influxdb with username/password

Telegraf Commands in This Lab

Command	Comment
`telegraf config`	Display configuration file contain
`telegraf --usage [inputs | outputs]`	Display sample config of a plugin
`telegraf --input-filter plugin1[:plugin2][...]`	Display input plugin configuration
`telegraf --output-filter plugin1[:plugin2][...]`	Display output plugin configuration

Lab List

1. Setting up Docker
2. Configuring Docker nonroot access and start on boot
3. Installing, configuring and running InfluxDB container
4. Installing Telegraf and fetching configuration file
5. Copy Telegraf configuration files to /etc/telegraf
6. Edit Telegraf configuration files
7. Running Telegraf

Lab List

8. Installing Grafana container and retrieving configuration file
9. Running Grafana
10. Adding data source and creating the first dashboard in Grafana
11. Adding Panels in the dashboard of Grafana
12. Setting up alert channel of Grafana
13. Upgrade Grafana to latest version
14. Configuring variables and template (optional)


LOGICALIS
Business and technology working as one
We are architects of change
Together we own the possible