

Basic Monitoring System in practical



About Speaker



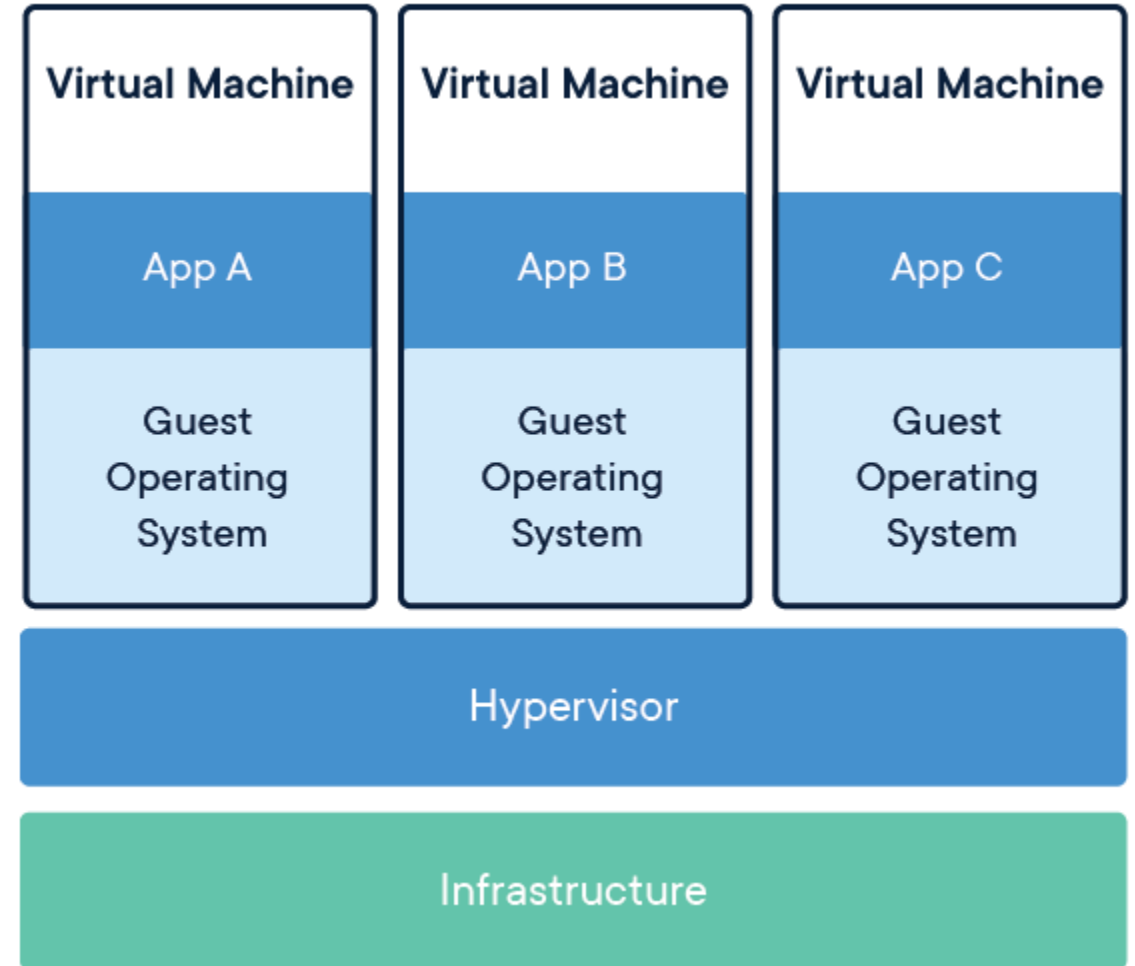
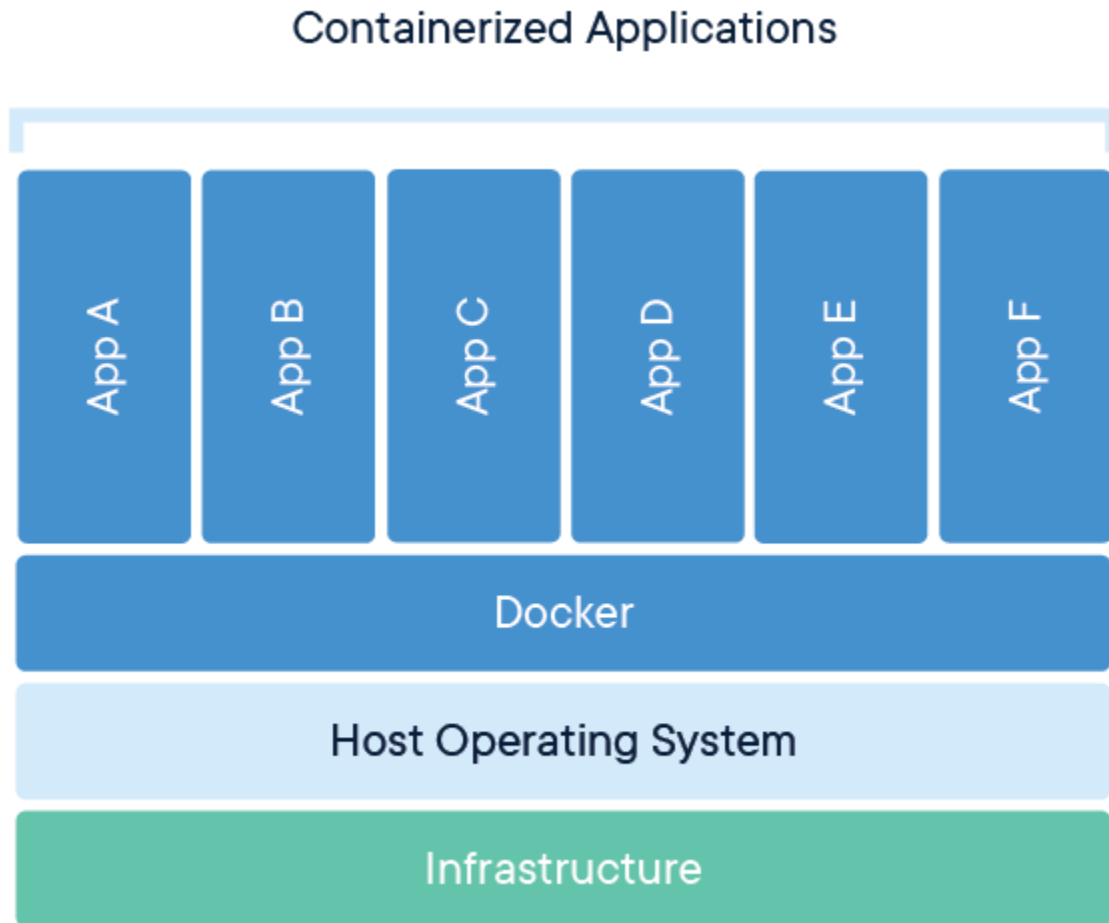
Mr. Kant Suwannasit DevOps Engineer

- 5 Years of experience in “Monitoring System”
- 3 Years in DevOps Engineer
- 2 Years in DBA-Ops Engineer
- 1 Year in Incident Management

Workshop Prerequisite : Docker

<https://www.docker.com/get-started>

ใครไม่มี Docker อาจไม่ได้ทำบาง workshop



Download Slide

<https://github.com/kantsuw/codemia1001>

Murphy's Law

**IF SOMETHING CAN
GO WRONG IT WILL**

MURPHYS LAW

Developer Nightmare



The worst nightmare

Try to restart service, but problem still there

.....

6 Hours passed, incident not resolved

.....

War room !!!

War room !!!



Lesson from the nightmare

How to know when something went wrong ?

.....

Without data, you're blind

.....

Who have data will win the war !!

and also sleep....

Agenda

Introduction to DevOps

Introduction to Monitoring System

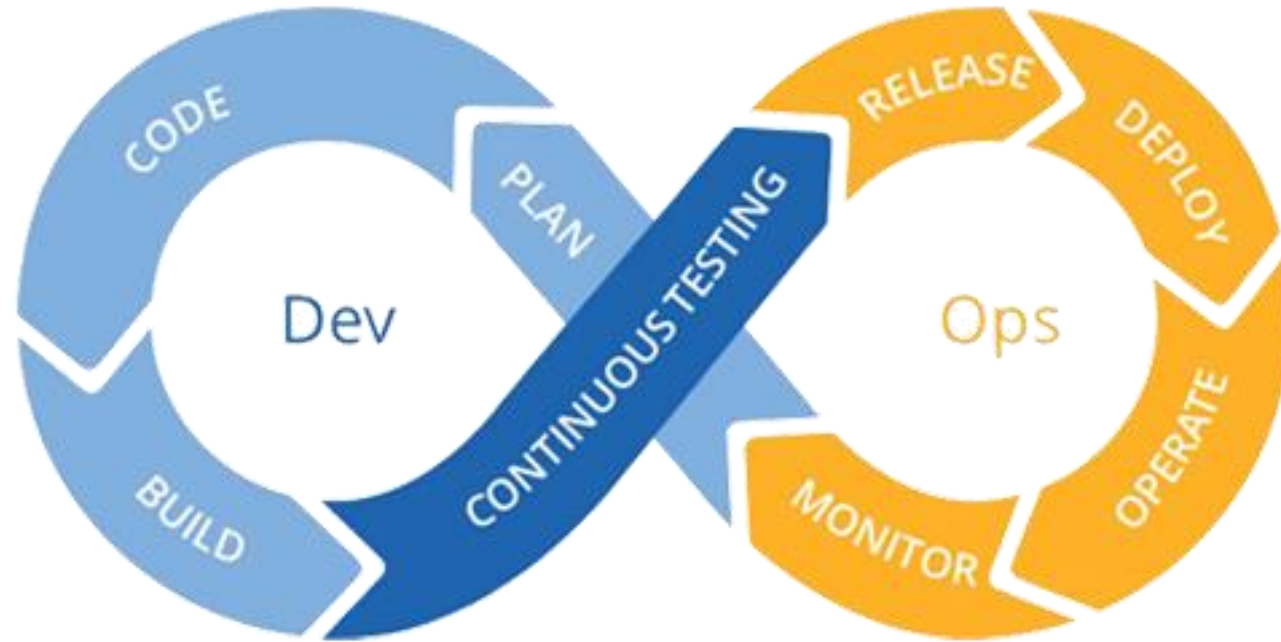
- What is Monitoring System ?
- Core Concept
- Monitoring Components
 - Workshop 1 - 7

Automate Problem Detection

- Prerequisite & Baseline
- Alerting & Processing
 - Workshop 8

DevOps

DevOps LifeCycle and Features





Why DevOps



Scalability

Flexibility

Automation

Reliability

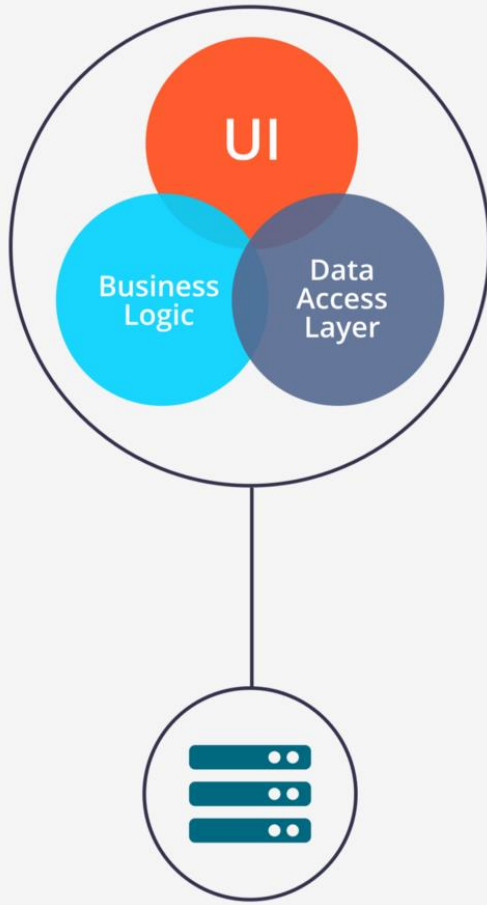




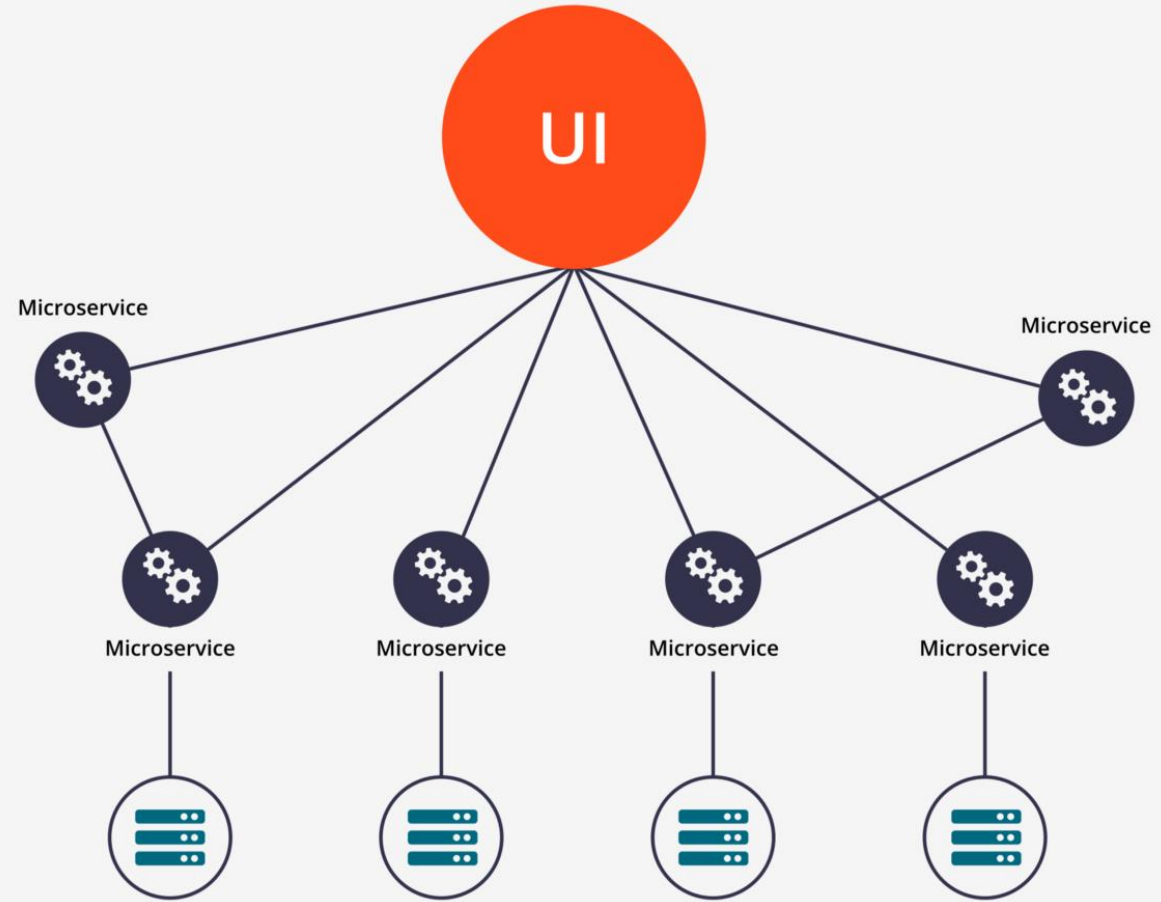
"In the new world,
it is not the big
fish which eats the
small fish, it's the
fast fish which
eats the **slow fish**"

Klaus Schwab
Founder and Executive Chairman
World Economic Forum

Monolith VS Microservice



Monolithic Architecture



Microservice Architecture

Introduction to Monitoring System

What is Monitoring System

Monitoring is the tools and processes which you can measure and manage your systems

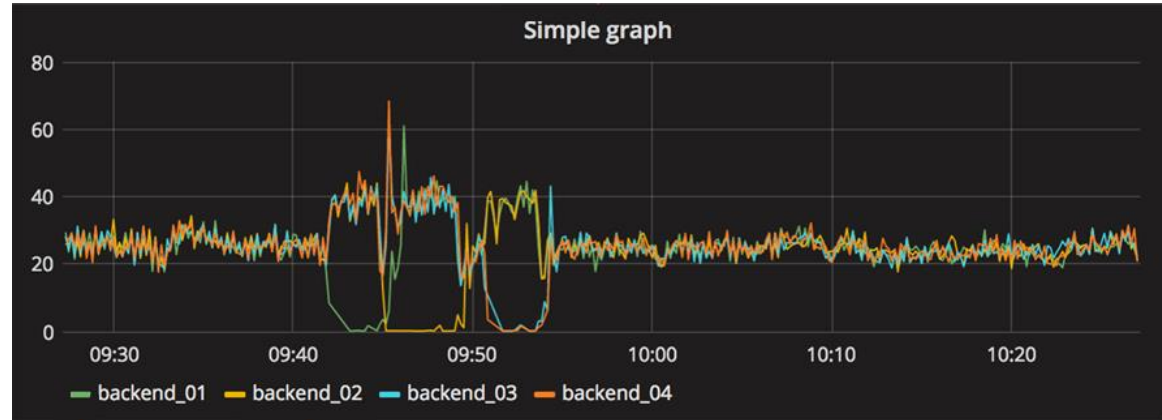
- Business value
- Information Technology
- Processes
- Everything



Why do we need Monitoring

1. Analysis in real time

2. Problem Detection



3. Incident Management

4. Feedback , Planning





“Data can tell you what opinions to hear; and conversely, opinions can tell you what data to read.”

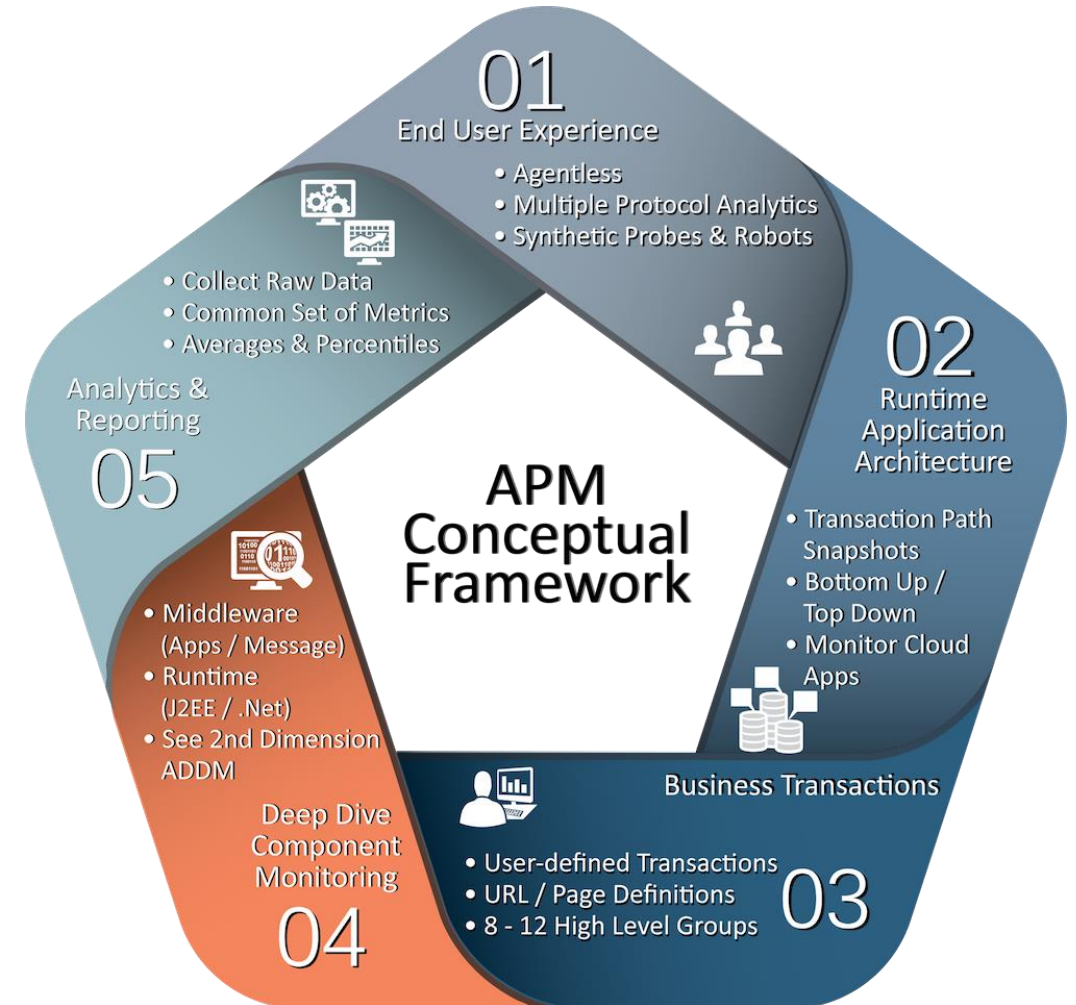
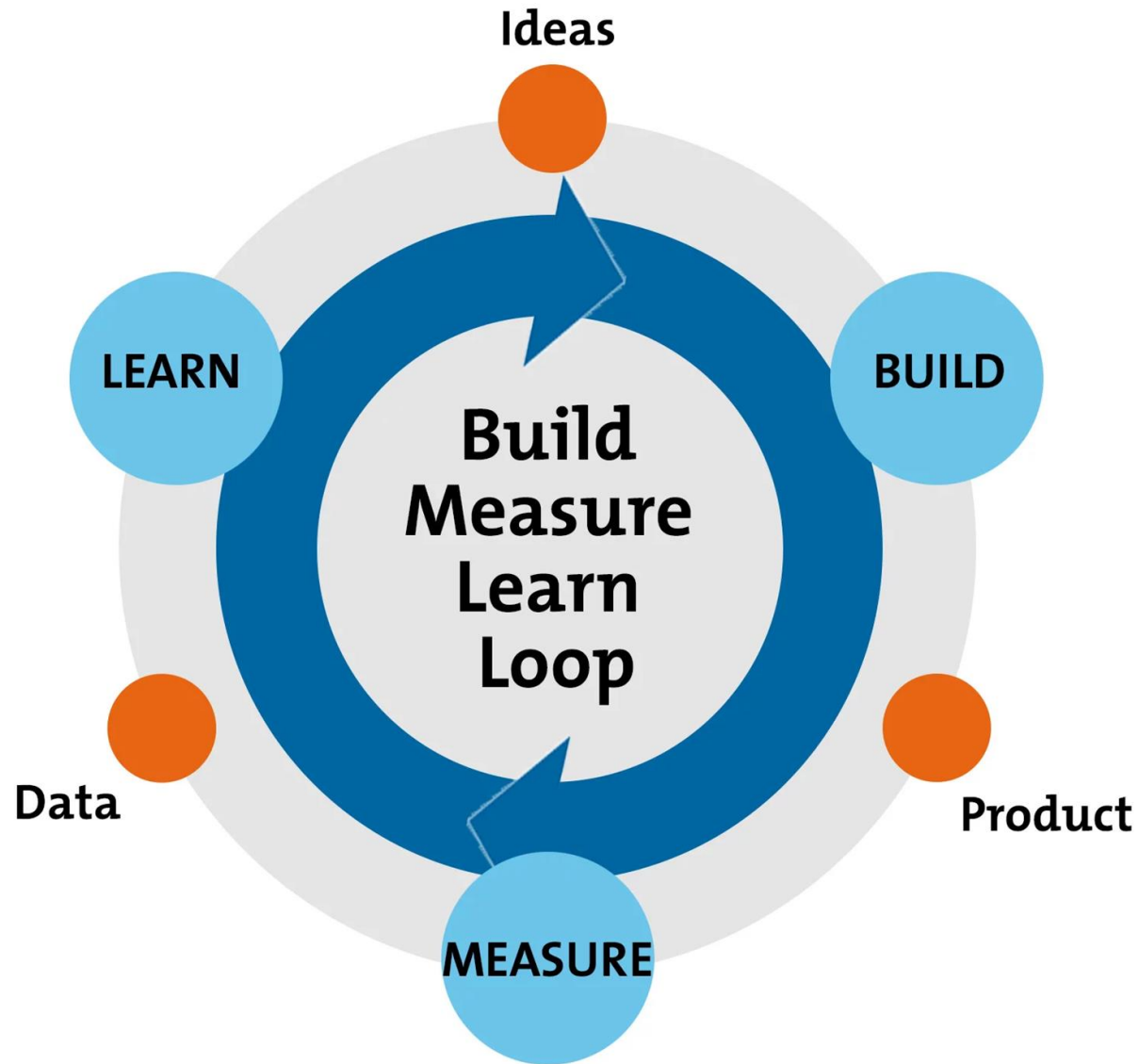
J.T. Trollman, @jtroll at @Facebook

**In God we
trust, all
others bring
data.**

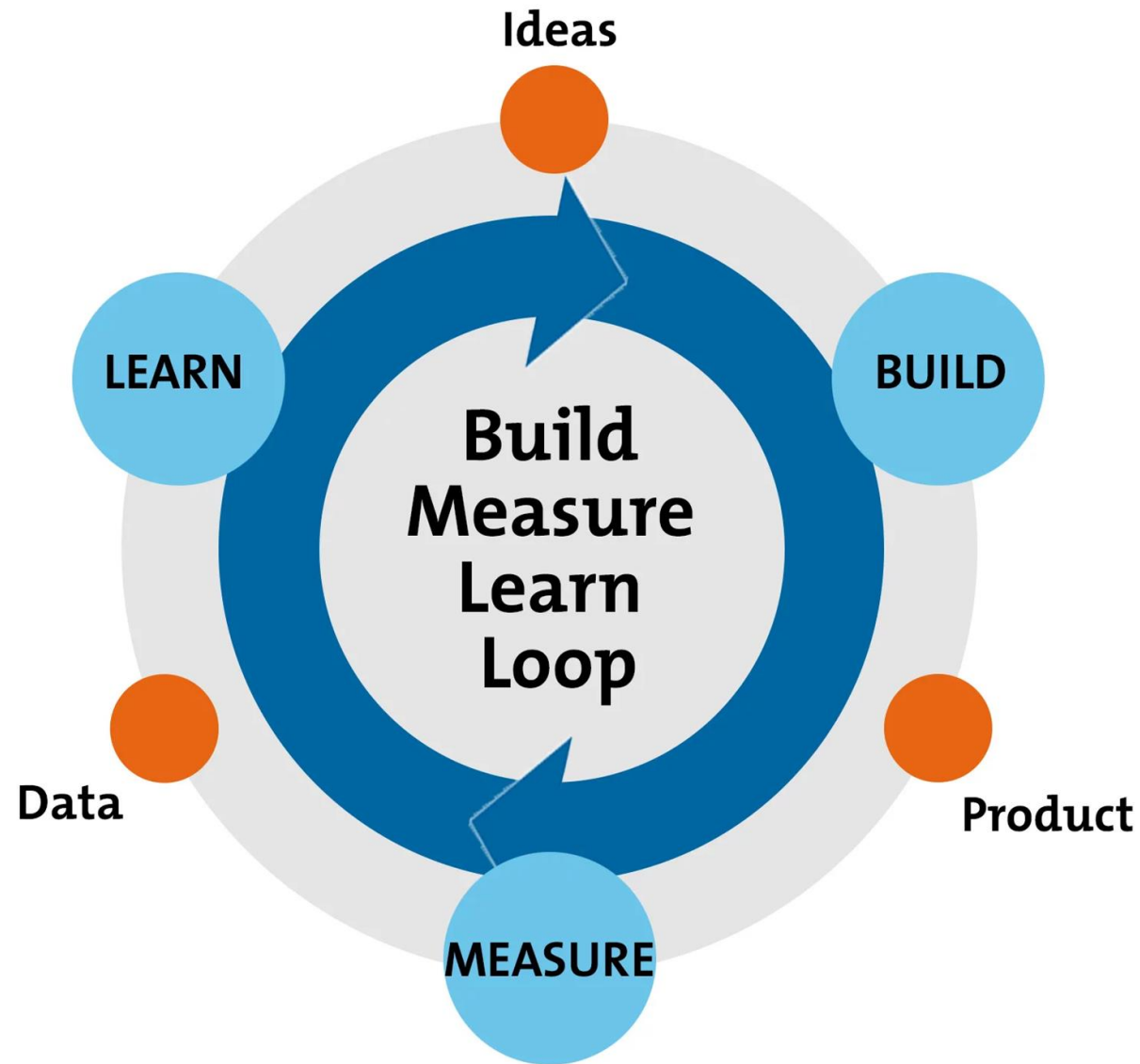
–William E. Deming



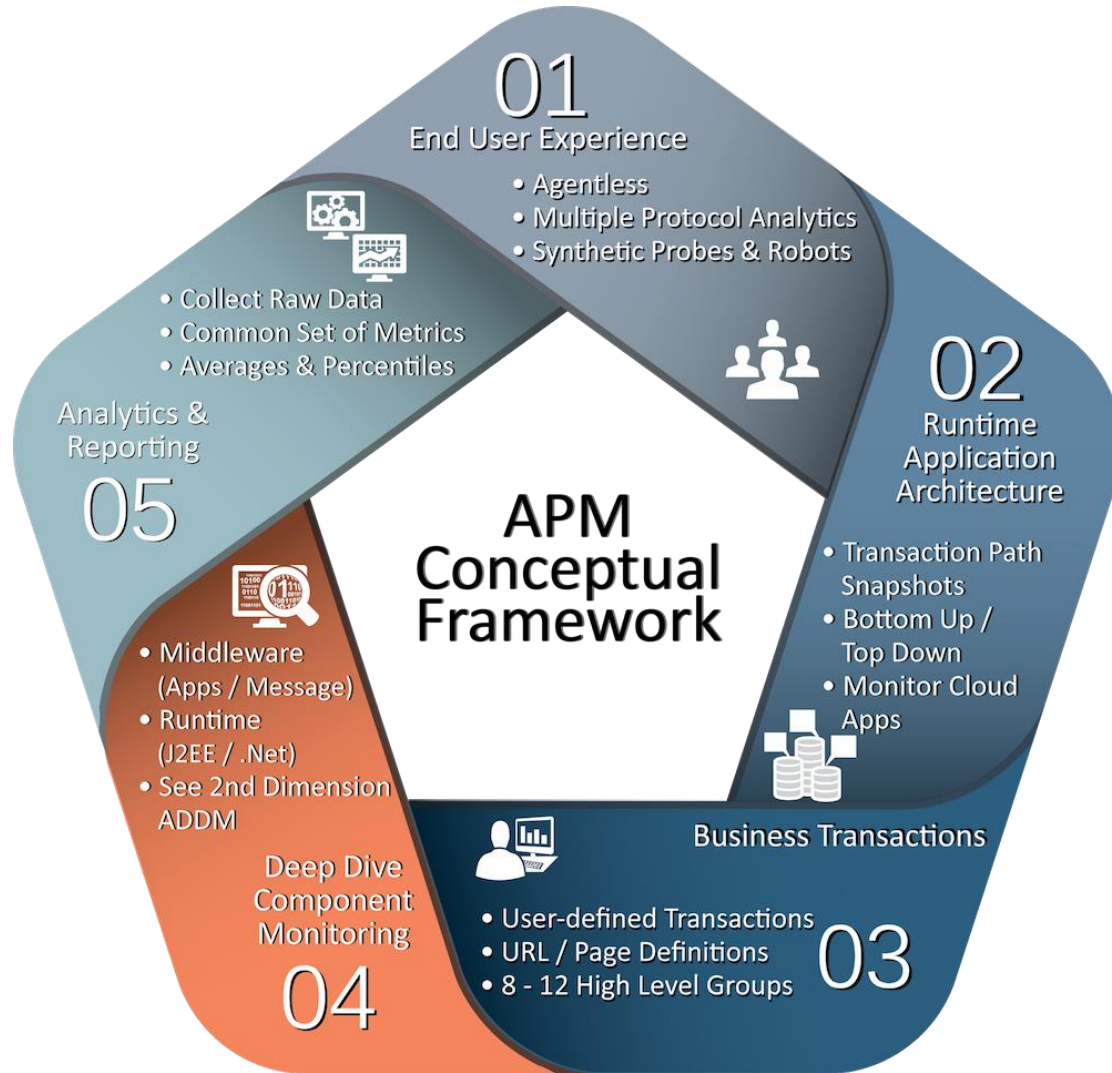
Core Concepts



Measurement Driven Development (MDD)

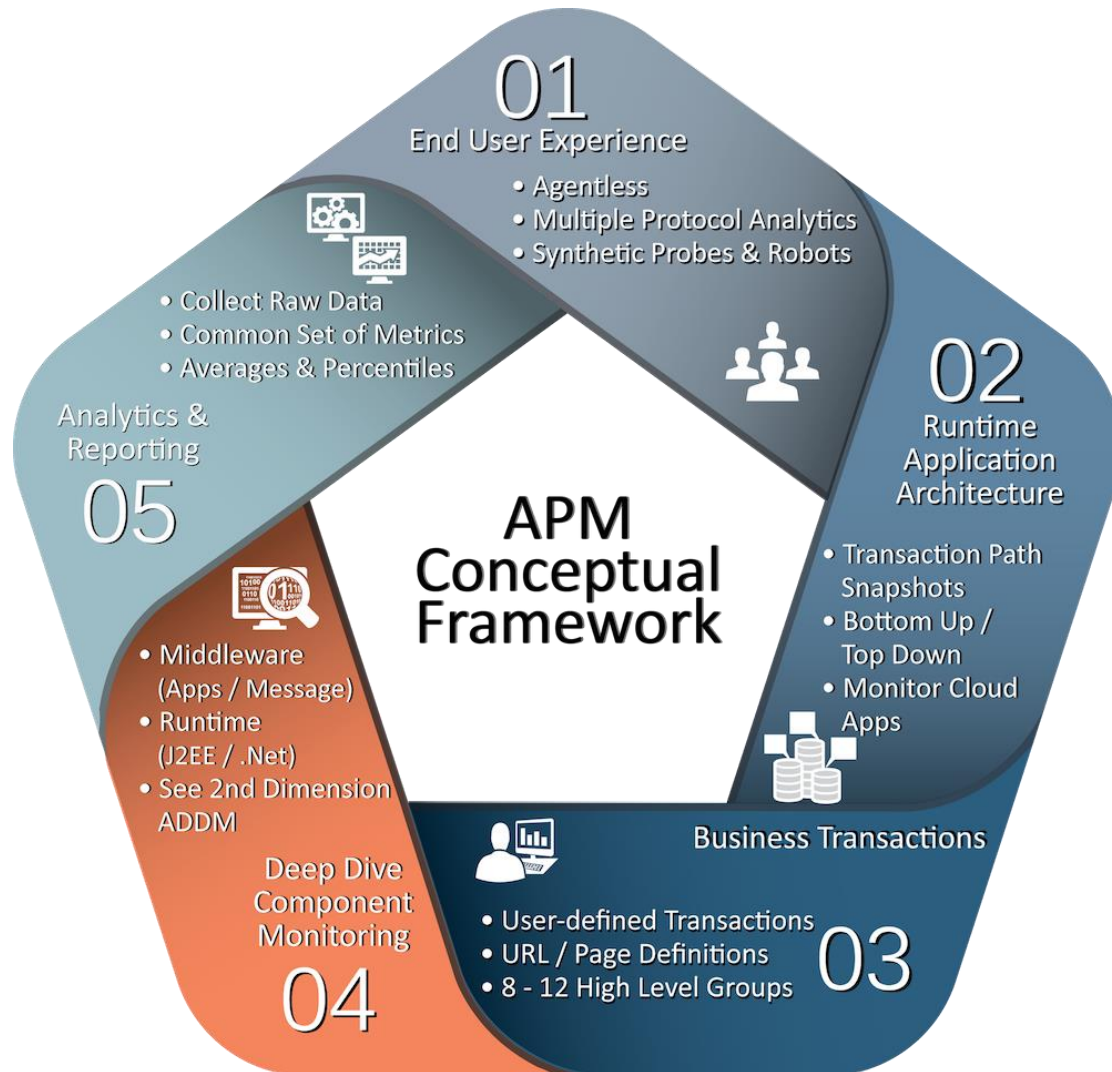


Application Performance Monitoring (1)



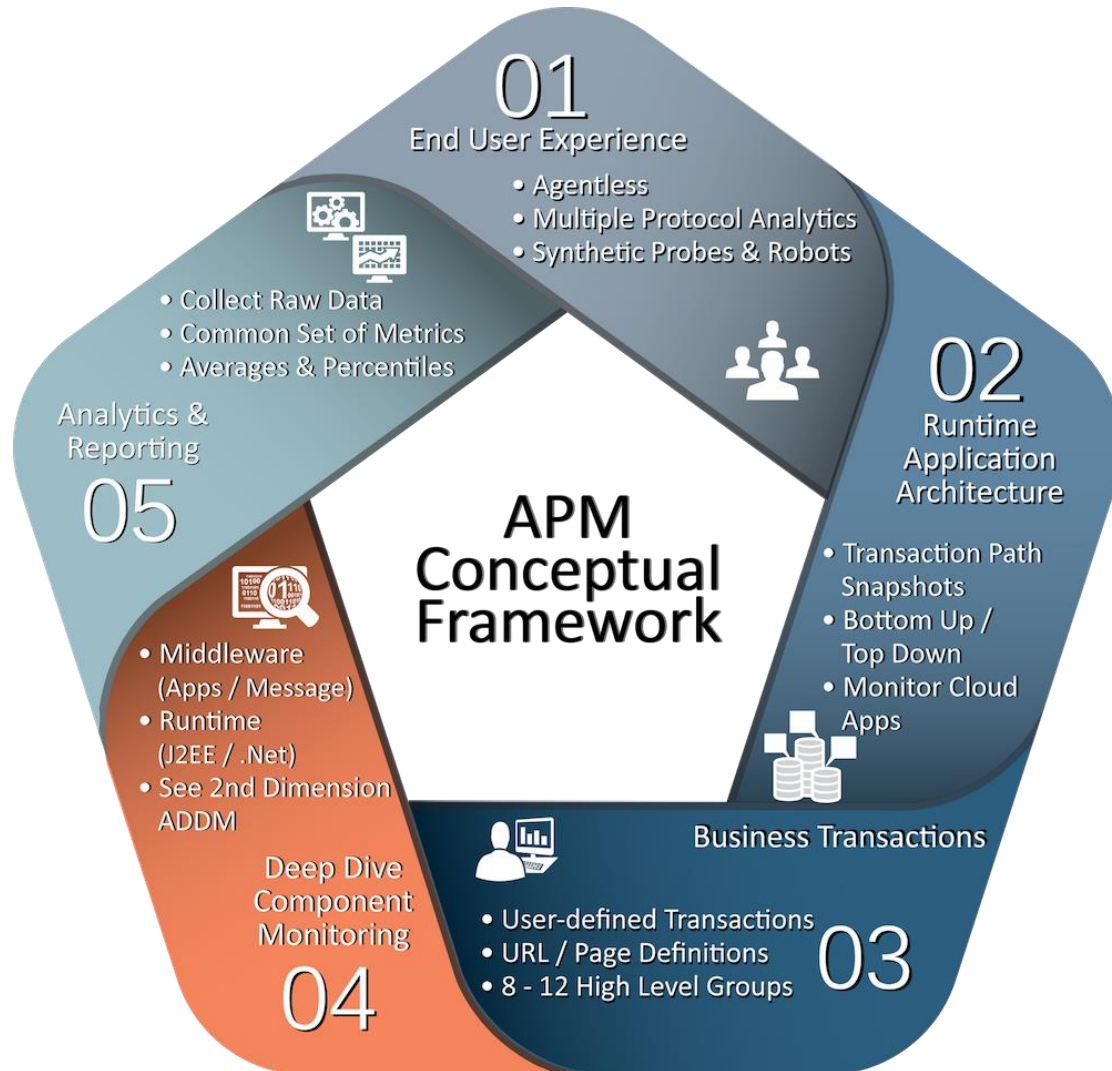
- End User Experience
 - Page load time
 - Runtime Error
 - Clicks & Conversion
 - etc.

Application Performance Monitoring (2)



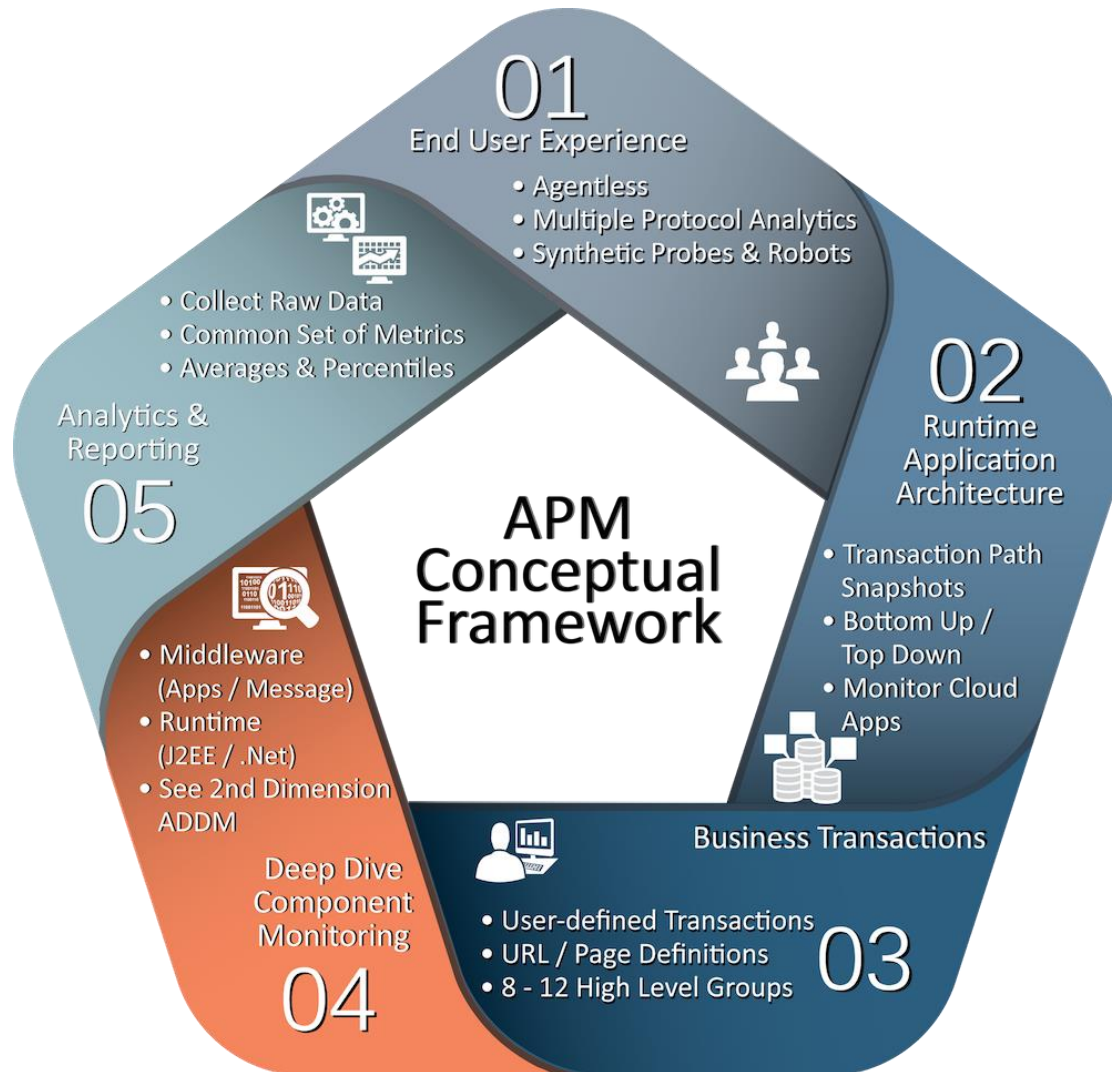
- Runtime Application Architecture
 - Application Server(s)
 - APIs
 - Database
 - Networks
 - Caching

Application Performance Monitoring (3)



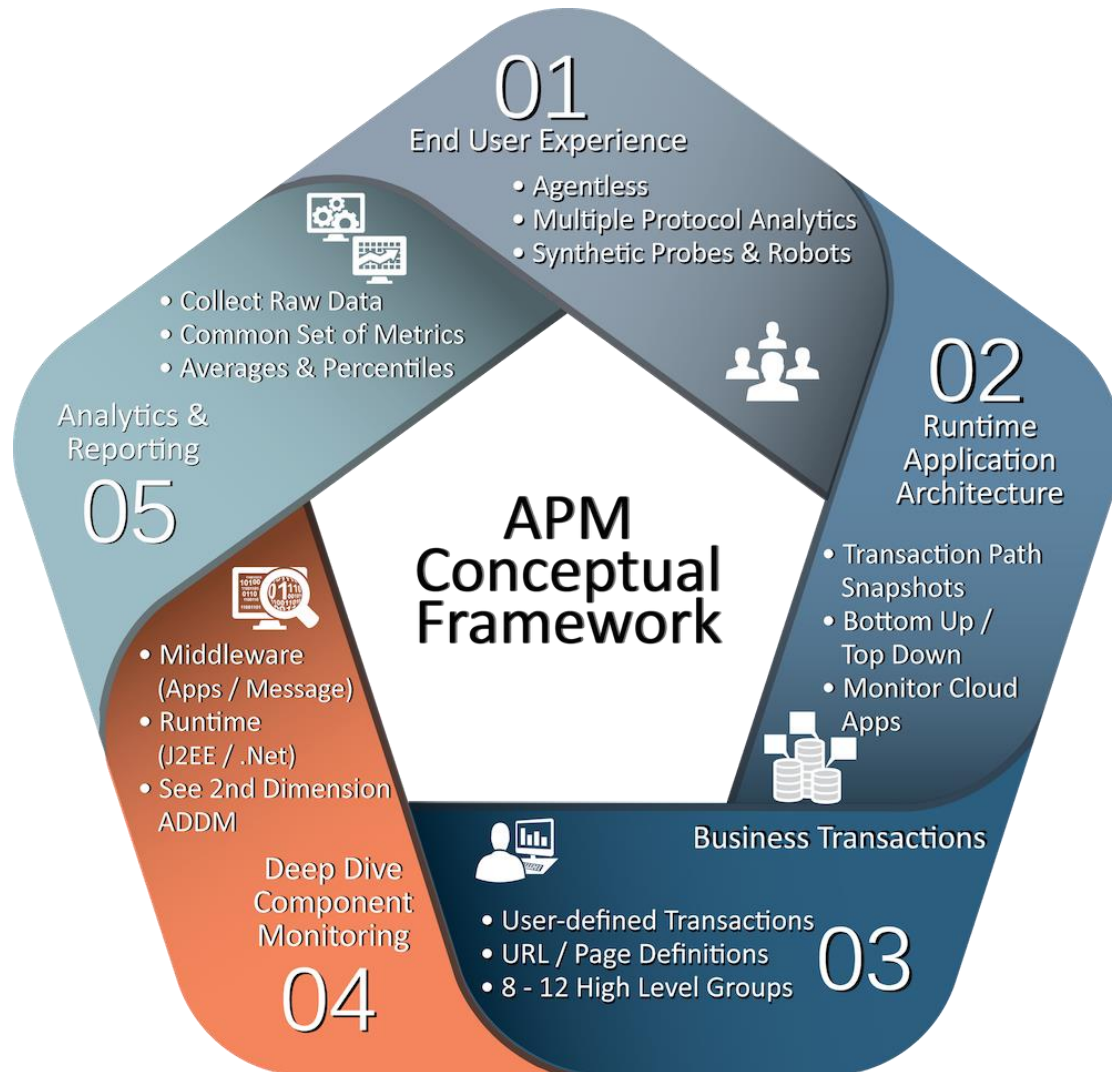
- Business Transaction
 - Buy/Sell
 - Cancel
 - Claim
 - Rewards Programs

Application Performance Monitoring (4)



- Deep Drive Component Monitoring
 - Middleware
 - Dependencies service
 - Third party

Application Performance Monitoring (5)



- Analytics & Report
 - Main funnels
 - Conversion rates
 - SLAs

Monitoring Components

1. Data (Measurement)



2. Collector



3. Storage

4. Visualization tools



Data



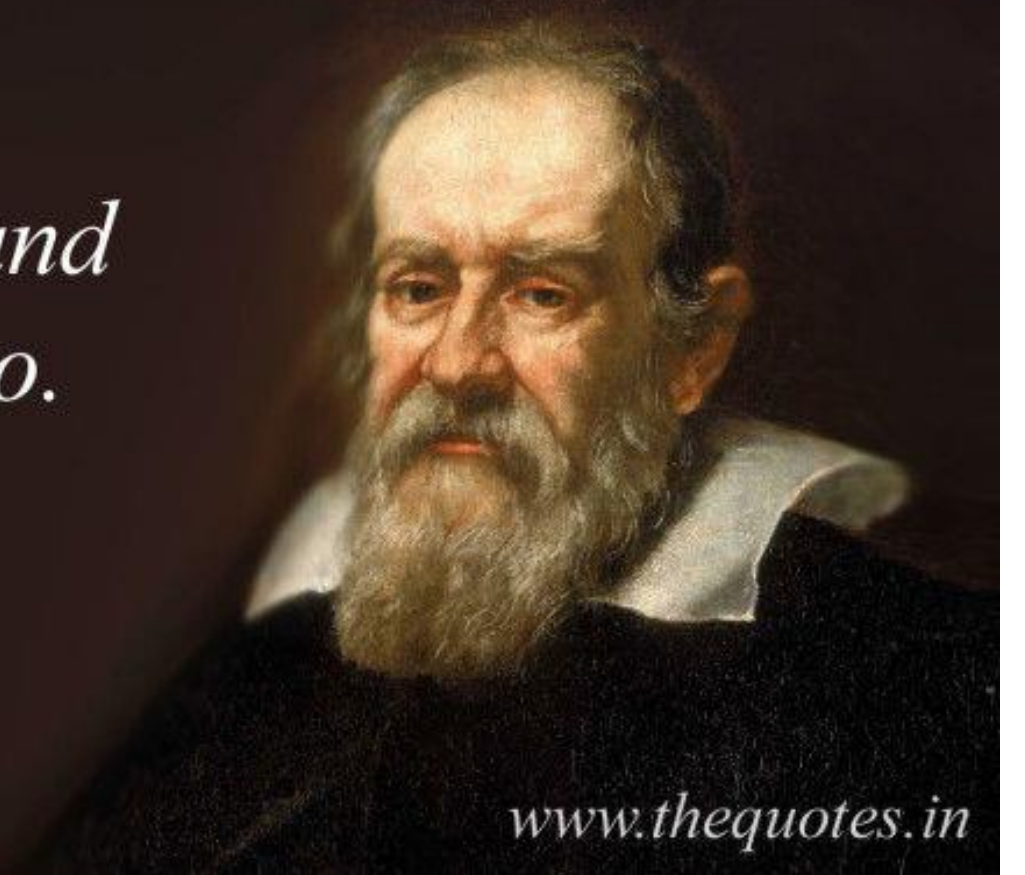
DATA

**IS THE MOST ABUNDANT
ELEMENT IN THE
NONPROFIT SECTOR**

Data

*Measure what is measurable, and
make measurable what is not so.*

Galileo Galilei



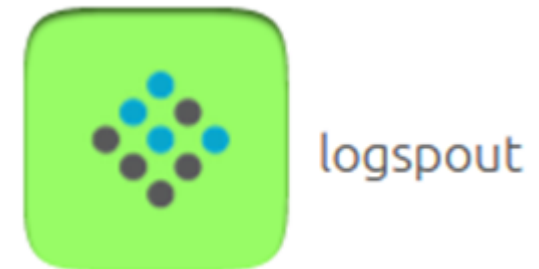
www.thequotes.in

Collector

Metrics metrics are a **measurement** at a **point in time** for the system it **repletely**

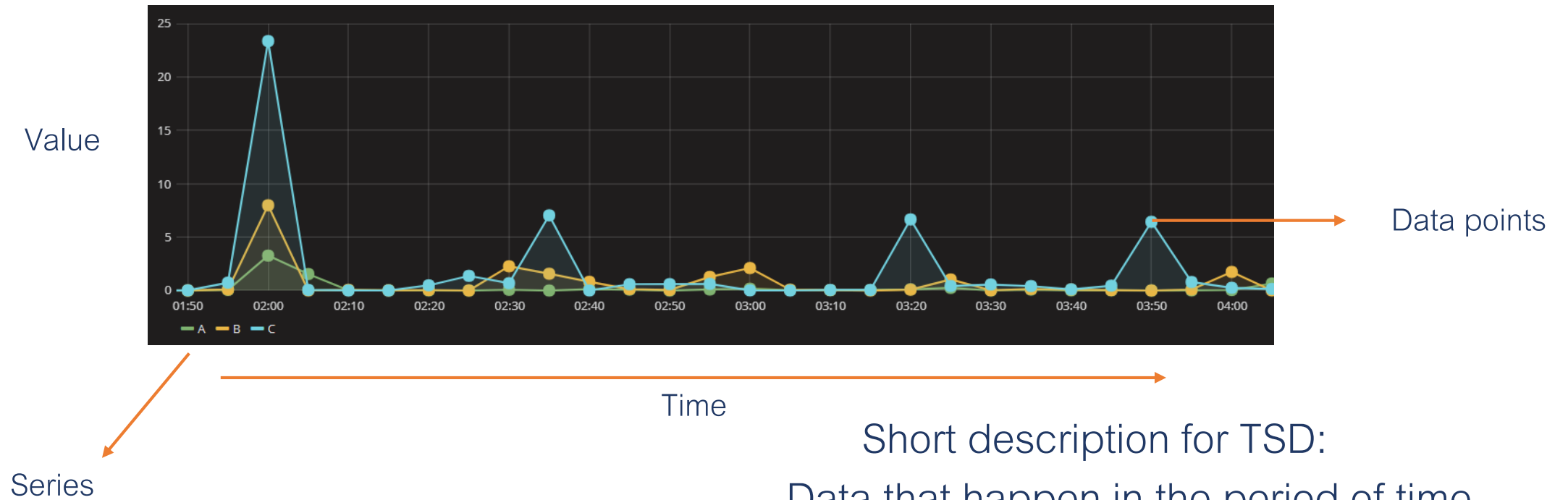


Logs logs are **messages** that **system** generated when an **event** has **happened** to describe the event



Logs may be collected **any time** an **event happened** , but **metrics** are **collected** at **fixed-time intervals**

Storage – Time Series Databases



Elasticsearch

- Written in Java - No cross-platform issue
 - An open-source “Search engine”
 - Build on top of Apache Lucene
- Document store with **JSON**
 - Every field is **indexed** and **searchable**
- Source for download
 - <https://www.elastic.co/downloads/elasticsearch>

MAC

- brew update
- brew install elasticsearch

Windows

- <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.4.msi>
- Easy install with UI

Elasticsearch – Keyword , Data

Term	Description
Cluster	Group of ElasticSearch nodes
Node	Instance of ElasticSearch
Index	Databases
Type	Table
Document	Row (JSON)
Fields	Columns

Example of JSON :

```
{“name”: “Kant” , “age” : 27 , “timestamp”: 20190622T14:00:00Z}
```

```
{“name”: “Bin” , “age” : 25 , “timestamp”: 20190617T15:00:00Z}
```

Data type :

- String

- Bigint

- Datetime

Workshop 1.0 Install Elasticsearch

<https://bit.ly/2ZEt6s5>

Workshop 1.1 Elasticsearch.yml

<https://bit.ly/2FkYoMF>

Workshop 1.2 restful-api, Cluster info, cat

Restful API : Web Service allow HTTP Method GET, POST, DELETE, UPDATE

```
← → ↻ 🏠 xezor.com:9200

{
  "name" : "HT94XAr",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "fWHOYbtFR0uF15Kltz-6Cw",
  "version" : {
    "number" : "6.2.2",
    "build_hash" : "10b1edd",
    "build_date" : "2018-02-16T19:01:30.685723Z",
    "build_snapshot" : false,
    "lucene_version" : "7.2.1",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

```
← → ↻ 🏠 xezor.com:9200/_cat/

=^.=
/_cat/allocation
/_cat/shards
/_cat/shards/{index}
/_cat/master
/_cat/nodes
/_cat/tasks
/_cat/indices
/_cat/indices/{index}
/_cat/segments
/_cat/segments/{index}
/_cat/count
/_cat/count/{index}
/_cat/recovery
/_cat/recovery/{index}
/_cat/health
/_cat/pending_tasks
/_cat/aliases
/_cat/aliases/{alias}
/_cat/thread_pool
/_cat/thread_pool/{thread_pools}
/_cat/plugins
/_cat/fielddata
/_cat/fielddata/{fields}
/_cat/nodeattrs
/_cat/repositories
/_cat/snapshots/{repository}
/_cat/templates
```

<https://bit.ly/2L9EBUv>

Workshop 1.3 Create Index , insert data , search

- GET : Search data
- POST : Create Index , Insert data



Postman

A screenshot of a Postman GET request. The URL bar shows 'http://13.251.6.15:9200/bnk48/_search'. The 'Authorization' tab is selected, showing 'No Auth'. The 'Body' tab is also visible, showing a JSON response in 'Pretty' format. The response is a search result for 'Nink'.

```
1 {
2   "took": 2,
3   "timed_out": false,
4   "shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 7228,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "bnk48",
16        "_type": "twitter_follower",
17        "_id": "uiqAA2IBnKrIb-LqfLpj",
18        "_score": 1,
19        "_source": {
20          "BnkName": "Nink",
21          "Followers": 256,
22          "Timestamp": 1520477174000
23        }
24      }
25    ]
26  }
27 }
```

A screenshot of a Postman POST request. The URL bar shows 'http://xezor.com:9200/bnk/bnk'. The 'Body' tab is selected, showing a JSON request in 'Pretty' format. The request is to create a new index 'bnk'.

```
1 {
2   "_index": "bnk",
3   "_type": "bnk",
4   "_id": "v0dUL2IBD_BOXUzWuoGq",
5   "_version": 1,
6   "result": "created",
7   "shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  },
12  "_seq_no": 0,
13  "_primary_term": 1
14 }
```

<https://bit.ly/2XYqAfV>

Workshop 1.3 Create Index , insert data , search

Create Index :

Post : url : localhost:9200/test/test

body :

```
{
  "name": "your_name",
  "age": 30,
  "timestamp": "2018-04-28T14:30:00Z"
}
```

```
1 = {
2   "_index": "test",
3   "_type": "test",
4   "_id": "WntDCGMBeamp-RSSFotm",
5   "_version": 1,
6   "result": "created",
7   "_shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  },
12   "_seq_no": 0,
13   "_primary_term": 1
14 }
```

Insert more rows :

Post : url : localhost:9200/test/test

body :

```
{
  "name": "your_friend_name",
  "age": 30,
  "timestamp": "2018-04-28T14:31:00Z"
}
```

```
1 = {
2   "_index": "test",
3   "_type": "test",
4   "_id": "W3tDCGMBeamp-RSSu4st",
5   "_version": 1,
6   "result": "created",
7   "_shards": {
8     "total": 2,
9     "successful": 1,
10    "failed": 0
11  },
12   "_seq_no": 0,
13   "_primary_term": 1
14 }
```

Workshop 1.4 Update data , delete data and index

PUT : Update data to specific id

DELETE : Delete data, Delete Index

Update rows :

Put : url : localhost:9200/test/test/U3stCGMBeamp-RSSI4vm

body :

```
{  
  "name": "your_friend_name",  
  "age": 40, <- New age  
  "timestamp": "2018-04-28T14:35:00Z"  
  <- New time  
}
```

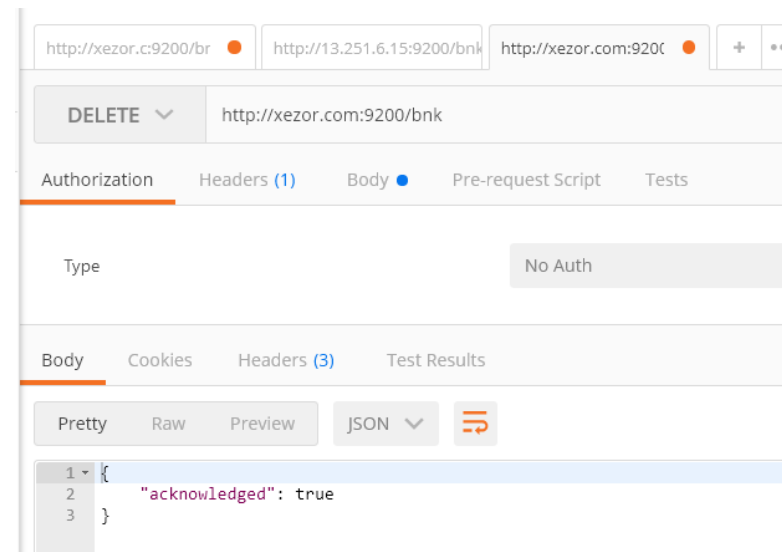
```
1 = {  
2   "_index": "test",  
3   "_type": "test",  
4   "_id": "U3stCGMBeamp-RSSI4vm",  
5   "_version": 2,  
6   "result": "updated",  
7   "_shards": {  
8     "total": 2,  
9     "successful": 1,  
10    "failed": 0  
11  },  
12  "_seq_no": 4,  
13  "_primary_term": 1  
14 }
```

Delete rows :

Delete : url : localhost:9200/test/test/U3stCGMBeamp-RSSI4vm

Delete Index :

Delete : url : localhost:9200/test/



<https://bit.ly/2x3CCJ5>

Workshop 1.5 basic query condition

GET : localhost:9200/info

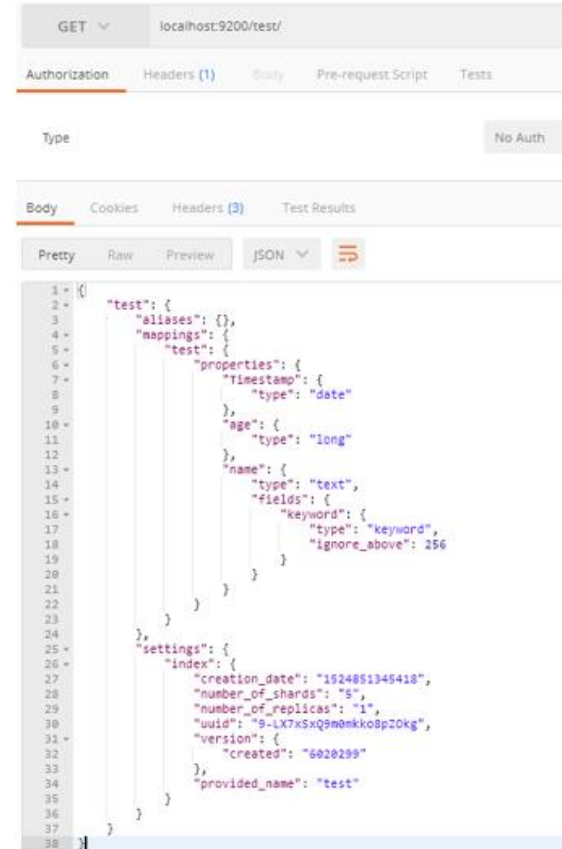
GET : localhost:9200/info/_search

GET : localhost:9200/info/_search?&size=10000

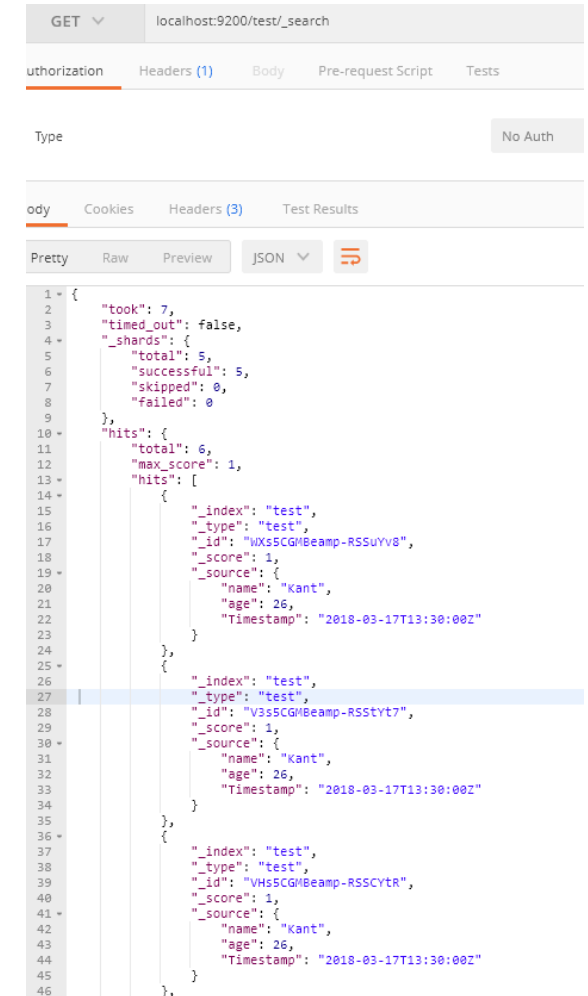
GET : localhost:9200/info/_search?q=name:Kant

GET : localhost:9200/info/info/_search?q=name:Kant

<https://bit.ly/2XYBdiH>



```
1 {
2   "test": {
3     "aliases": {},
4     "mappings": {
5       "test": {
6         "properties": {
7           "timestamp": {
8             "type": "date"
9           },
10          "age": {
11            "type": "long"
12          },
13          "name": {
14            "type": "text",
15            "fields": {
16              "keyword": {
17                "type": "keyword",
18                "ignore_above": 256
19              }
20            }
21          }
22        }
23      },
24      "settings": {
25        "index": {
26          "creation_date": "1524851345418",
27          "number_of_shards": "5",
28          "number_of_replicas": "1",
29          "uuid": "9-LX7x5xQ9m0kko8p20kg",
30          "version": {
31            "created": "6020299"
32          },
33          "provided_name": "test"
34        }
35      }
36    }
37  }
38 }
```



```
1 {
2   "took": 7,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 6,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "test",
16        "_type": "test",
17        "_id": "VHs5CGMBeamp-RSSuYv8",
18        "_score": 1,
19        "_source": {
20          "name": "Kant",
21          "age": 26,
22          "timestamp": "2018-03-17T13:30:00Z"
23        }
24      },
25      {
26        "_index": "test",
27        "_type": "test",
28        "_id": "V3s5CGMBeamp-RSStyt7",
29        "_score": 1,
30        "_source": {
31          "name": "Kant",
32          "age": 26,
33          "timestamp": "2018-03-17T13:30:00Z"
34        }
35      },
36      {
37        "_index": "test",
38        "_type": "test",
39        "_id": "VHs5CGMBeamp-RSScytr",
40        "_score": 1,
41        "_source": {
42          "name": "Kant",
43          "age": 26,
44          "timestamp": "2018-03-17T13:30:00Z"
45        }
46      }
47    ]
48  }
49 }
```

<https://www.elastic.co/guide/en/elasticsearch/reference/current/search.html>

Workshop 1.6 Export, import data with nodejs

<https://bit.ly/2WO07jL>

Workshop 1.7 Example Application

<https://github.com/kantsuw/BXtoES>

Influxdb

- Written in Golang , can run in all of platform
- Value store database
 - Measurement Name , Tag , value , timestamp
- Source for download
 - <https://portal.influxdata.com/downloads/>

MAC

https://dl.influxdata.com/influxdb/releases/influxdb-1.7.6_darwin_amd64.tar.gz

tar zxvf influxdb-1.7.6_darwin_amd64.tar.gz

Windows

https://dl.influxdata.com/influxdb/releases/influxdb-1.7.6_windows_amd64.zip

[unzip influxdb-1.7.6_windows_amd64.zip](#)

Telegraf

- Written in Golang , can run in all of platform
- Collector that can collect many data type, many services
Ex. Server Metric , SQL , Redis , Mongo , Http-api
- Source for download
 - <https://portal.influxdata.com/downloads/>

MAC

```
brew update  
brew install telegraf
```

Windows

```
https://dl.influxdata.com/telegraf/releases/telegraf-1.11.0_windows_amd64.zip unzip  
telegraf-1.11.0_windows_amd64.zip
```

Workshop 2.0 Install Influxdb

<https://bit.ly/31Huo7t>

<https://bit.ly/2XXL9Ji>

Workshop 2.1 Create Database , Insert data

<https://bit.ly/2IZS52q>

Workshop 2.2 Install Telegraf + Conf

<https://bit.ly/2MTzOsy>

<https://bit.ly/31JR7zR>

<https://bit.ly/2WNOPRc>

<https://bit.ly/2WNP2Uu>

Workshop 2.3 Data from Telegraf in Influxdb

<https://bit.ly/2WTQmp2>

Prometheus

- Written in Golang , can run in all of platform
- Prometheus is storage and also collector by itself

Concept of pulling metrics

- Source for download

- <https://prometheus.io/download/>

MAC

brew update

brew install prometheus

Windows

<https://github.com/prometheus/prometheus/releases/download/v2.10.0/prometheus-2.10.0.windows-amd64.tar.gz>

telegraf-1.11.0_windows_amd64.zip

Workshop 3.1 Install Prometheus

<https://bit.ly/2L78XGO>

Workshop 3.2 Application Metric

<https://github.com/kantsuw/example-prometheus-nodejs>

<http://kantz.space:3001/metrics>

<http://kantz.space:9090/metrics>

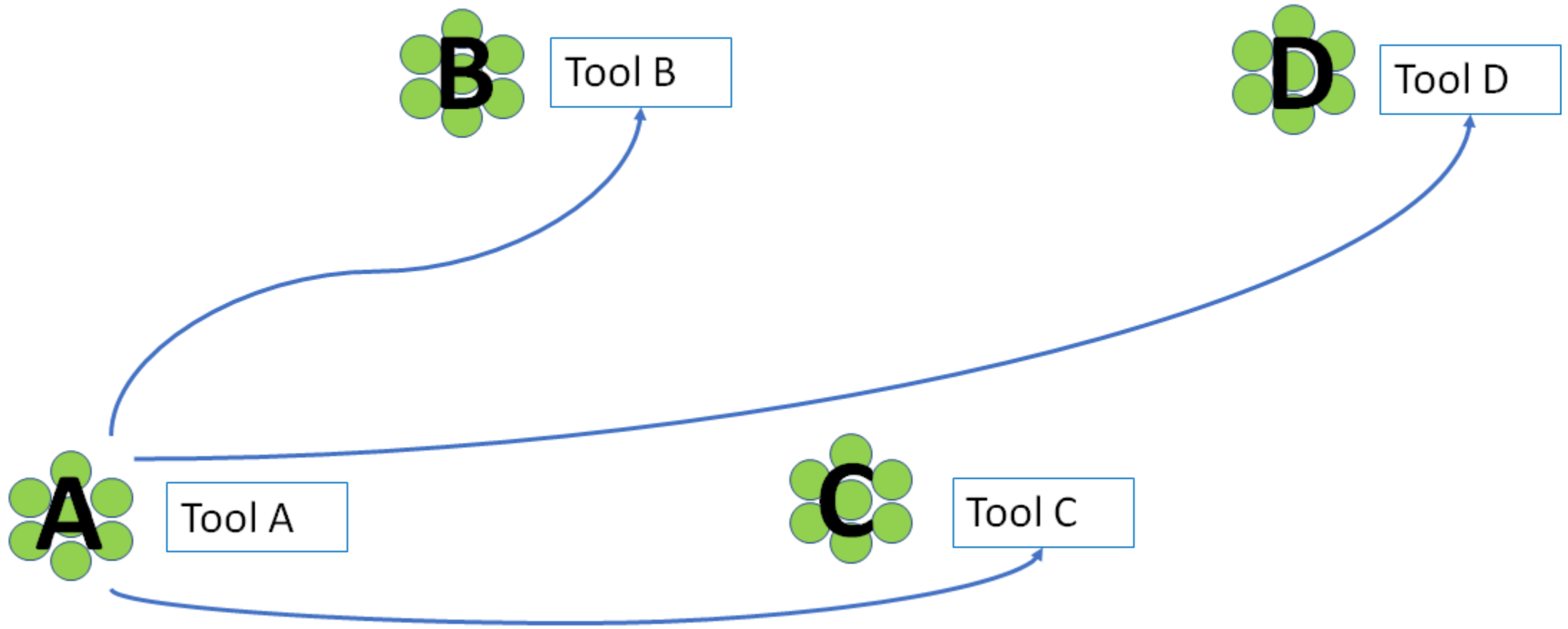
Workshop 3.3 Prometheus Configuration

<https://bit.ly/2XULkVO>

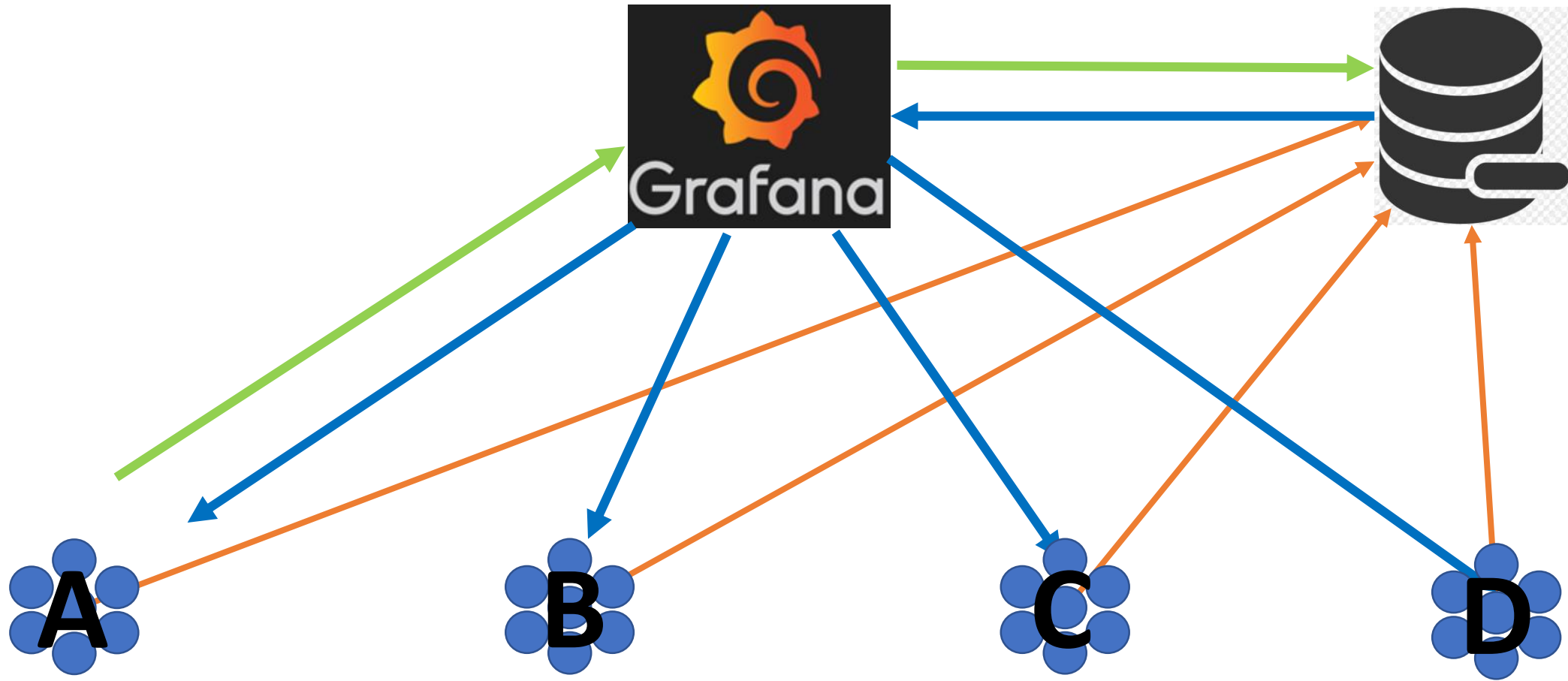
Workshop 3.4 Prometheus UI

<http://kantz.space:9090/graph>

Visualization tools - traditional



Visualization tools - Grafana



Grafana

- Written in Golang , can run in all of platform

Visualization Tool , not storage , not collector

- Source for download

<https://www.grafana.com/grafana/download>

- MAC

brew install grafana

brew services start grafana-server

- Windows

Unzip

Grafana-server.exe

- Grafana UI

<http://localhost:3000>

admin/admin

Workshop 4.1 Grafana Configuration

<https://bit.ly/2FkvZGP>

<https://bit.ly/2Zztk3G>

Workshop 4.2 Grafana create datasource

<http://localhost:3000/datasources>

<http://kantz.space:3000/datasources>

Workshop 4.3 Create graph with elasticsearch

<http://localhost:3000/dashboard/new?orgId=0>

<http://kantz.space:3000/dashboard/new?orgId=0>

Workshop 4.4 Create graph with influxdb

<http://localhost:3000/dashboard/new?orgId=0>

<http://kantz.space:3000/dashboard/new?orgId=0>

Workshop 4.5 Create graph with prometheus

<http://localhost:3000/dashboard/new?orgId=0>

<http://kantz.space:3000/dashboard/new?orgId=0>

Workshop 4.6 Access Control

<http://localhost:3000/admin/users>

<http://kantz.space:3000/admin/users>

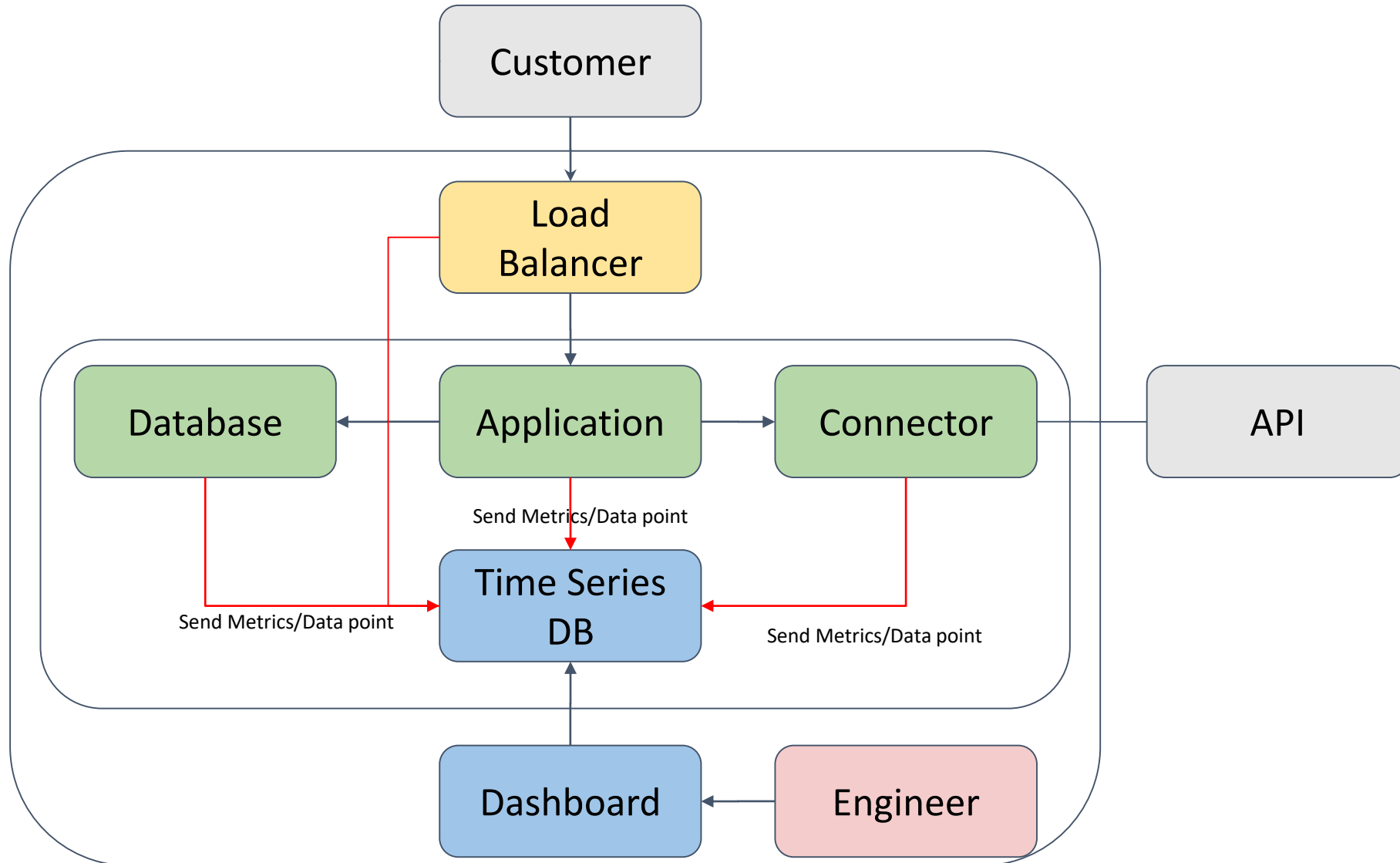
Workshop 4.6 Grafana Plugin

<https://grafana.com/plugins>

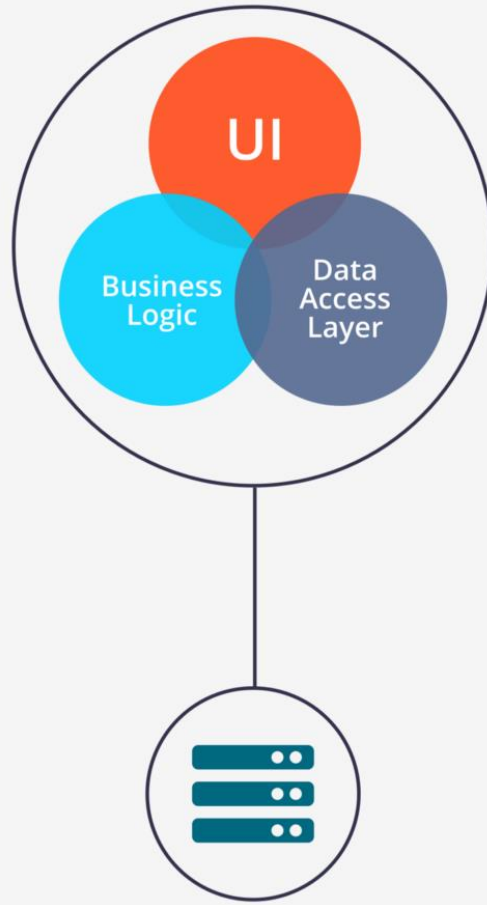
<https://grafana.com/plugins/grafana-clock-panel>

<https://grafana.com/plugins/grafana-piechart-panel>

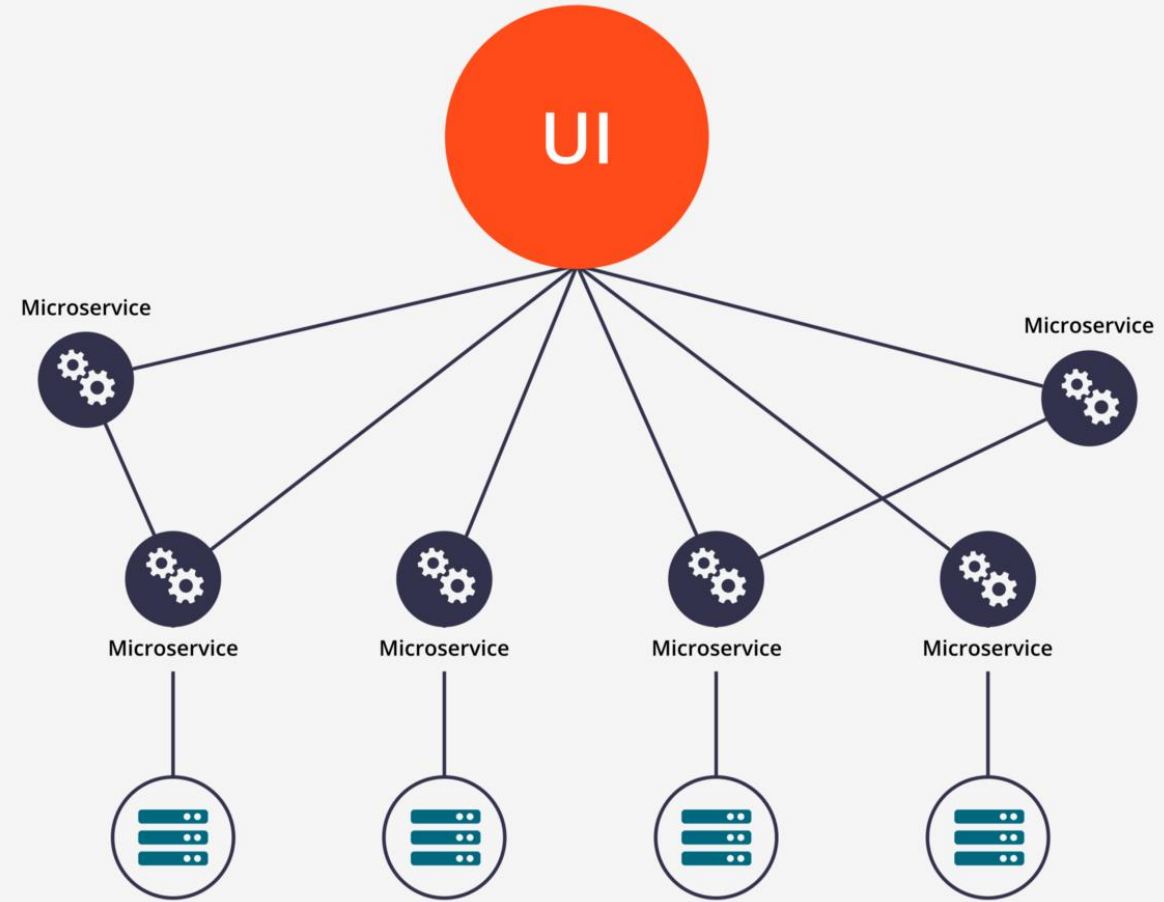
Sample Application Architecture



Monolith VS Microservice



Monolithic Architecture



Microservice Architecture

Showcase 1 Sample Monolith VS Microservice

https://github.com/kantsuw/selector_api

<https://github.com/kantsuw/BXtoES>

https://github.com/kantsuw/keeper_api

https://github.com/kantsuw/crypto_currency_api

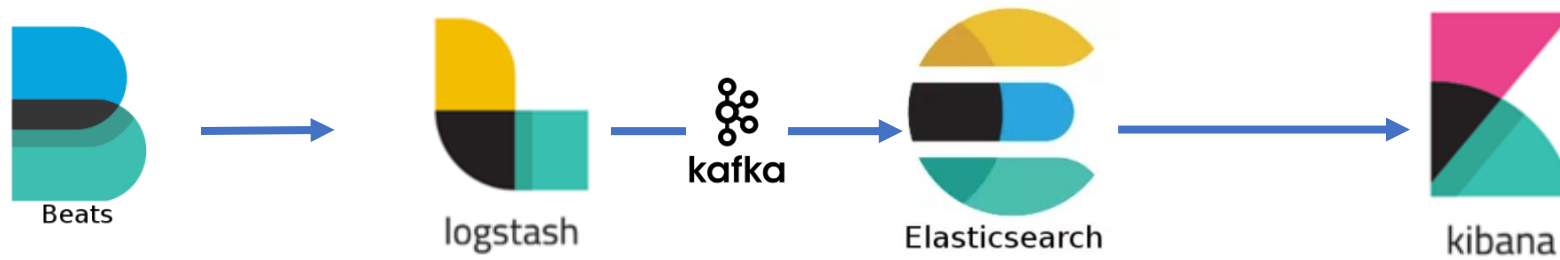
Showcase 2 CI/CD with TeamCity

<http://kantz.space:8111/overview>

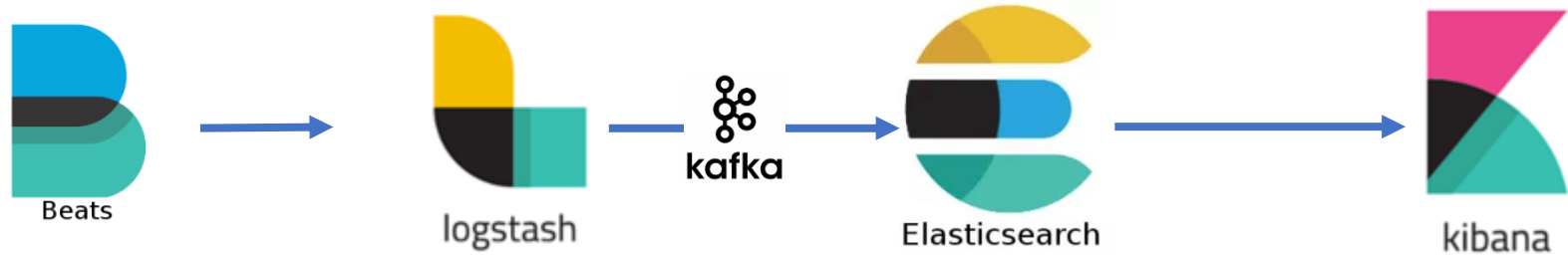
Showcase 3 Service Version Dashboard

Showcase 4 Sample Business Dashboard

Another stack : Beat Logstash Elasticsearch Kibana



Monitoring Stack

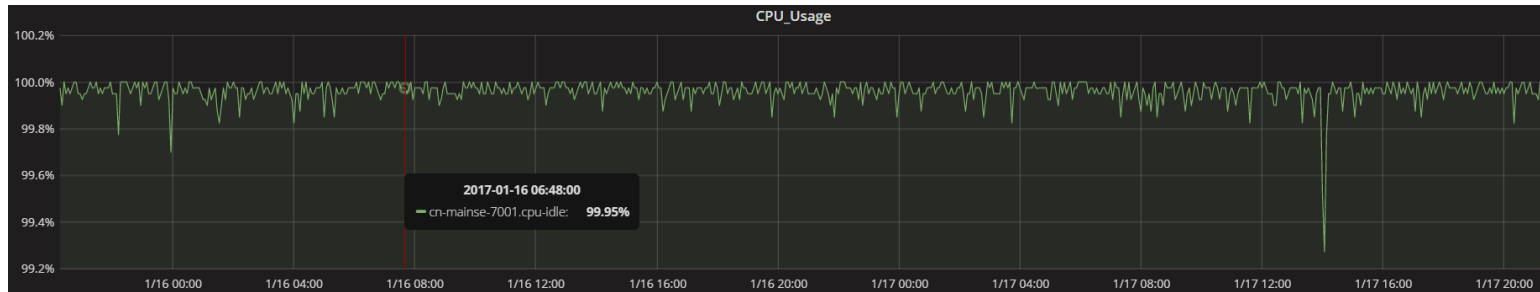


Automation Problem Detection

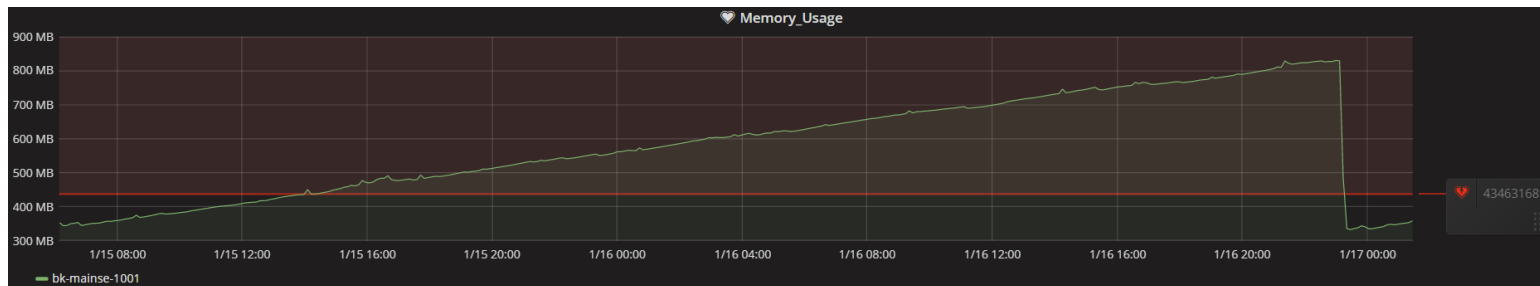
Prerequisite & Baseline

Prerequisite – What we should do before start monitoring,

1. Collecting the data to build data historical and statistics
2. Understanding nature of data

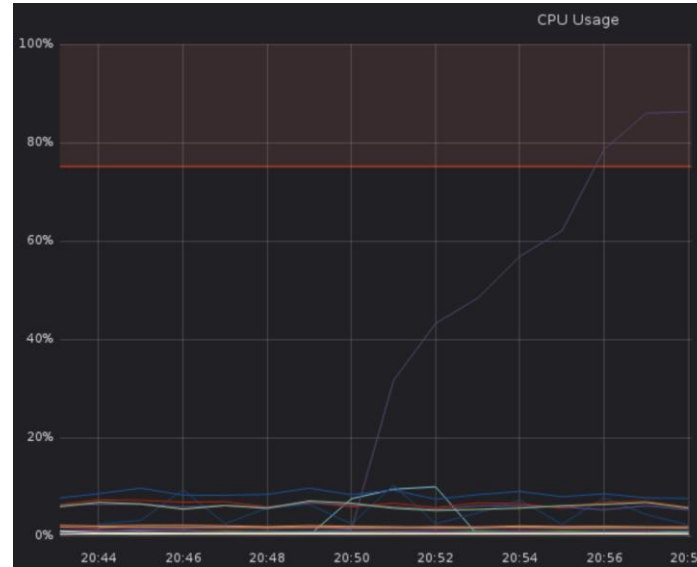


Baseline – A point or line that we used for comparisons



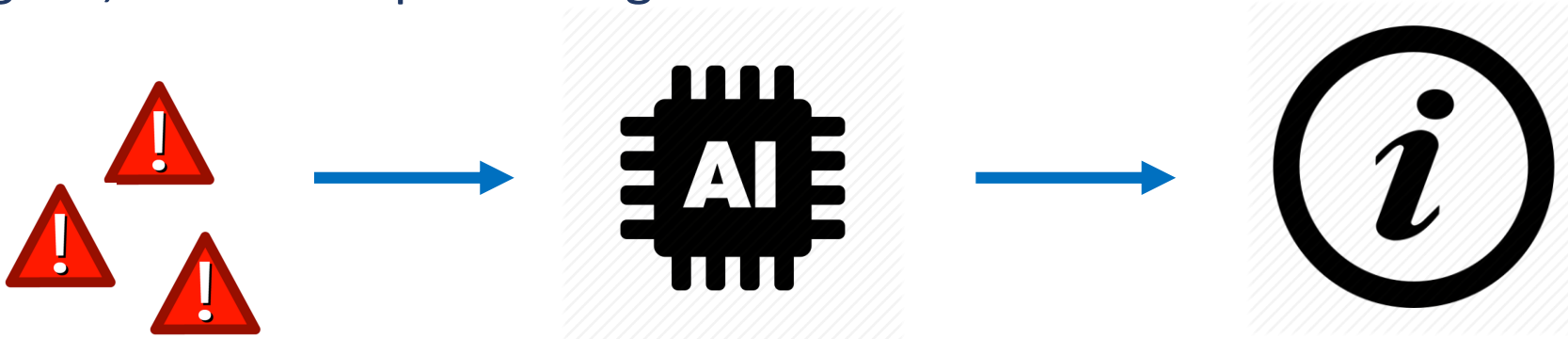
Alerting & Processing

When ever measurement go above baseline it will be alerting



Alerting can be receive by multi receivers

Let imagine , if we have processing that can correlation between alerts



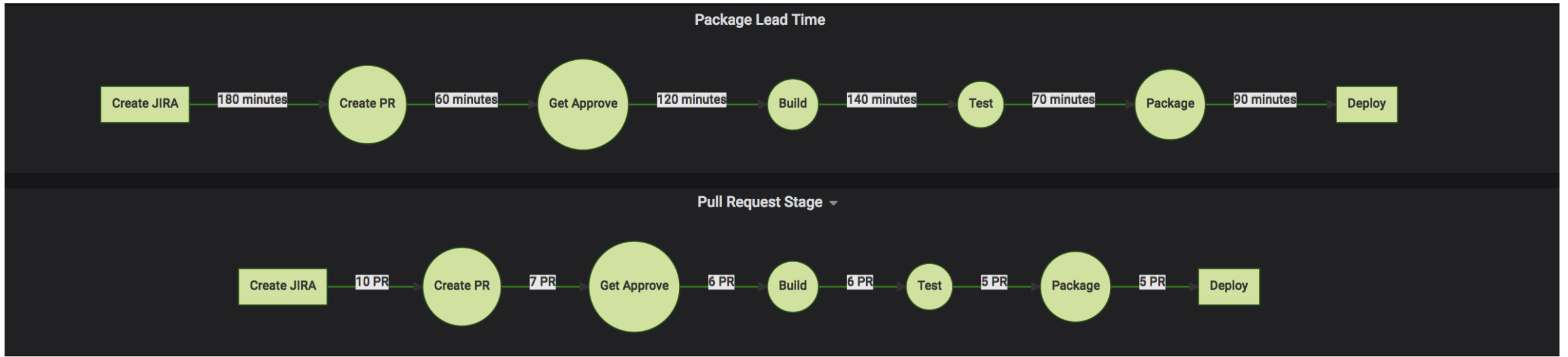
Workshop 5 Grafana Slack Notification

<http://kantz.space:3000/alerting/notifications>

<https://bit.ly/2N7XIAW>

Benefit of better monitoring (1)

Measuring Development process



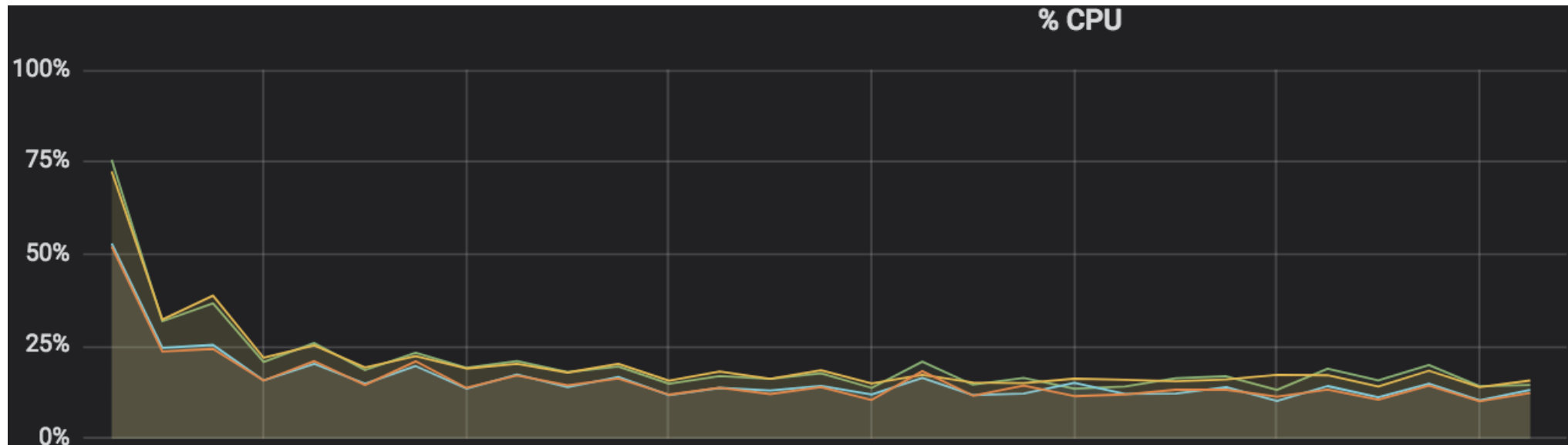
Reduce development time & improve productivity

Reduce 60% development time

Improve 370% productivity

Benefit of better monitoring (2)

Measuring operation performance



Improve Server Capacity & Reduce Operation Cost

Improve 60% of server capacity

Reduce 50% of Operation Cost

Benefit of better monitoring (3)

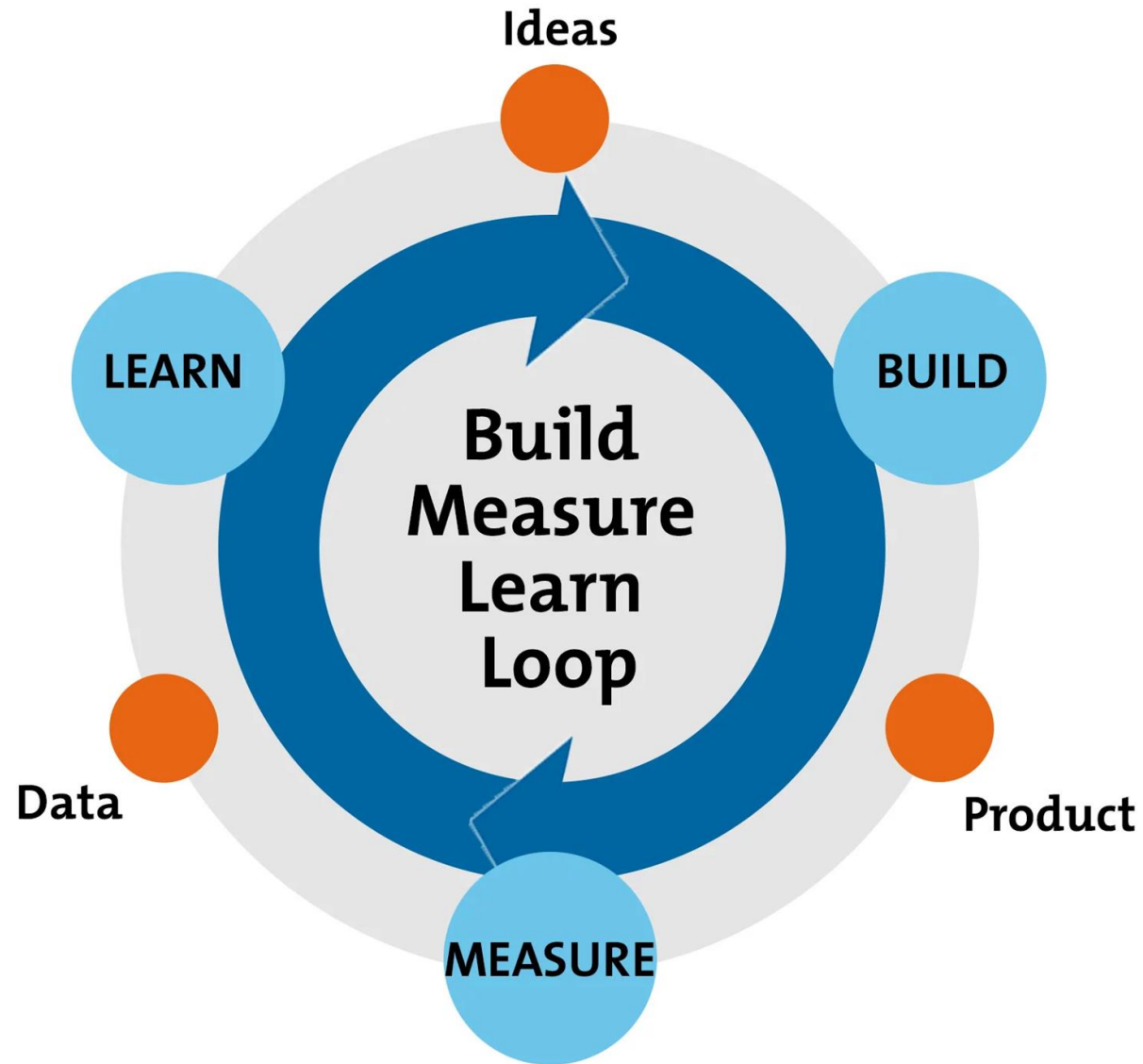
Measuring Code Quality

Period	Total Test	Test Passed	Test Failed	Test Passed (%)
Before-create	300	243	57	81.5
During-create	600	555	45	92.50
After-create	1000	980	20	99.80

Better Incident management

Period	Incident per week	Investigation time (day)	Down time (minutes) Per week	Availability (%)
Before-create	7	4.3	187	98.14
During-create	5	2.1	96	99.04
After-create	2	0.5	30	99.70

Summary



Special Thanks :

Varokas P.

Mahasak P.

Norbor (Nuboa)

Tirana J.

And Everyone

Q & A