

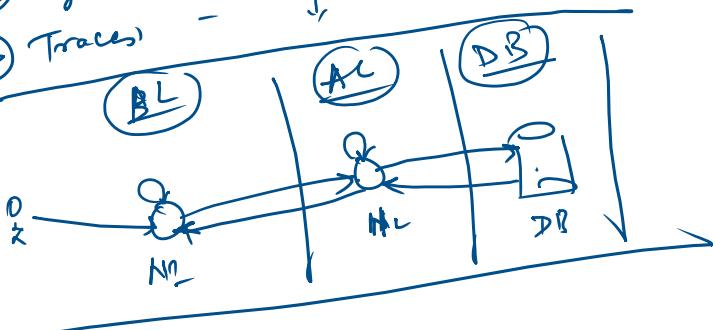
- ① License.
- ② Customer Support
- ③ Integration with other App.
- ④ Analysis Engine.

Monitoring
Receive

Observability

Pro active.

- ① Metrics - CPU = 80%
- ② logs - Timestamp
- ③ Traces



- ① Log Monitoring
- ② Infra Monitoring
- ③ King Monitoring
- ⑥ Cloud monitoring
- Dynatrace

- (2) Infra
 - (3) Container monitoring
 - (4) Application Monitoring
 - (5) Database Visibility,
- Dynatrace

- (1) Dynatrace
- (2) New Relic
- (3) App dynamics
- (4) Datadog

Dynatrace - In 2014,

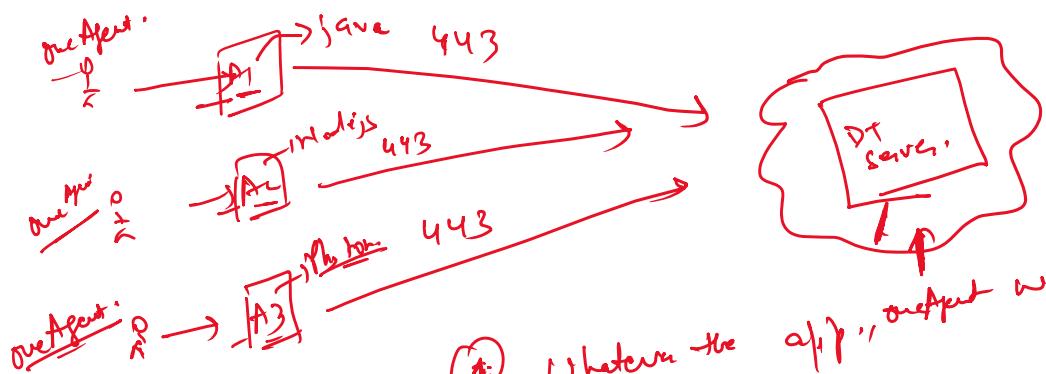
- (1) Dynatrace APM (Application Monitoring)
- (2) DUEM (Dynatrace user Experience management)
- (3) Synthetic Monitoring
- (4) Dynatrace Data Centre (DCRUM)
- (5) Dynatrace APM for mobile.

Ruixit, combined all the app. to the one App.

Dynatrace (APM tool)
↓
(Application Performance Management Tool)

- PT :-
- (1) Managed - Infra + App
 - (2) SaaS - App

② SaaS - App



- ① Whether the app's output will send the data to the DT server.
- ② Only dependency is the O.S (Windows / Linux)

① nginx - Reverse Proxy Server

① nginx - Reverse Proxy Server - Serve engine when the proxy sis.

② Elastic Search - Search engine when the proxy sis.

③ Cassandra/Hypercube - Distributed Database.

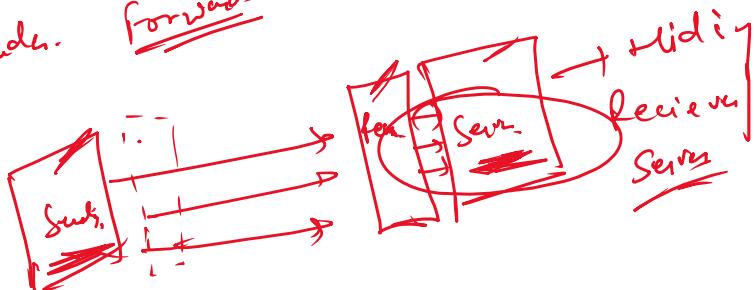
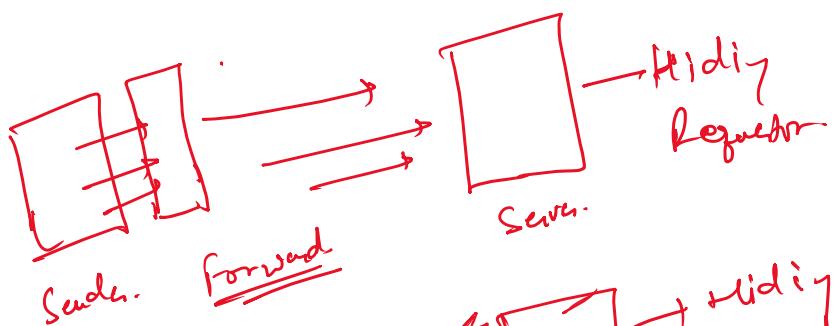
④ DT Server - collecting.

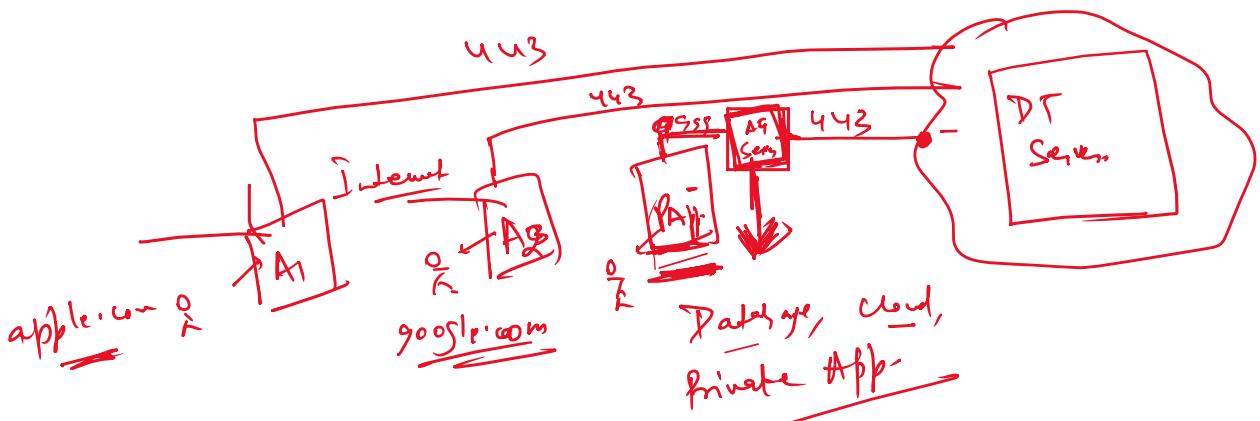
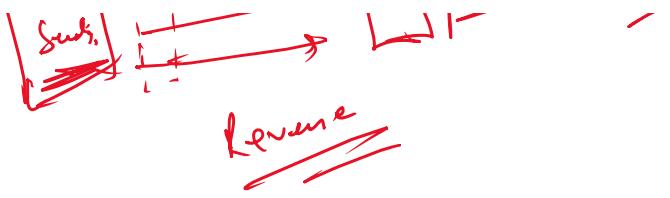
⑤ Active gate - light weight agent - DB, cloud mainly, private App

Proxy + hide the info

① Forward

② Reverse



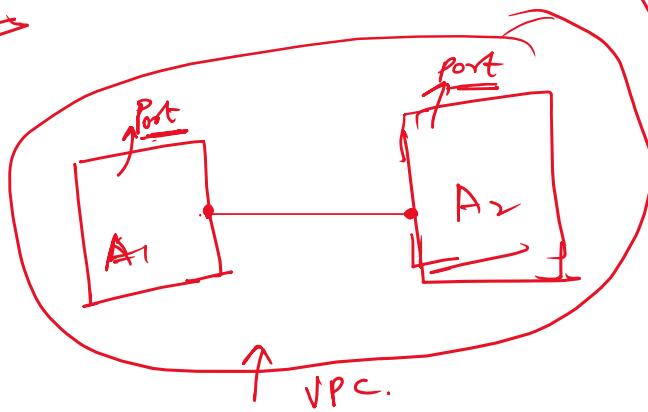


SaaS -

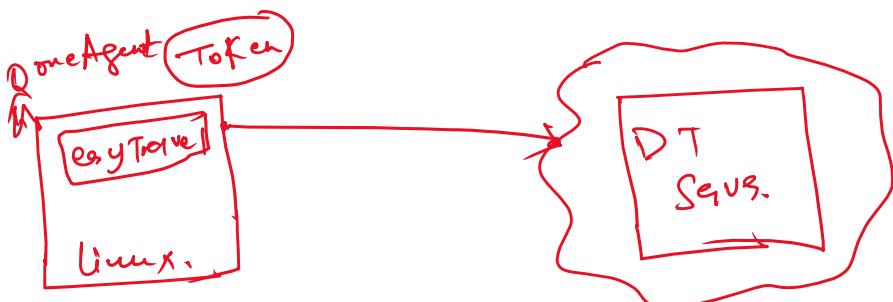
- ① No Infra.
- ② No firewall
- ③ No VPC
- ④ API

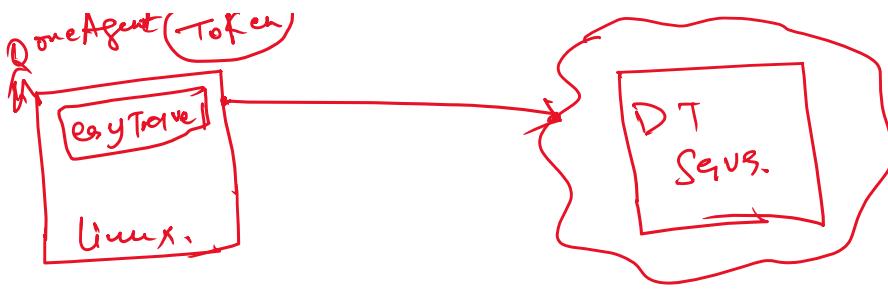
Managed

- ① Infra.
- ② Firewall
- ③ VPC
- ④ API



① DT Server
② Application (easyTravel - Java & .Net)



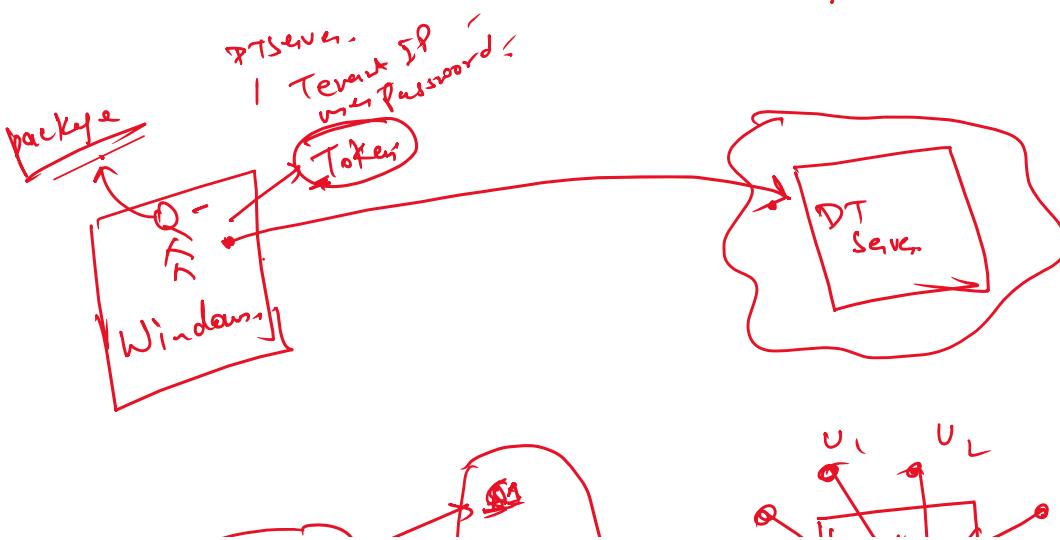
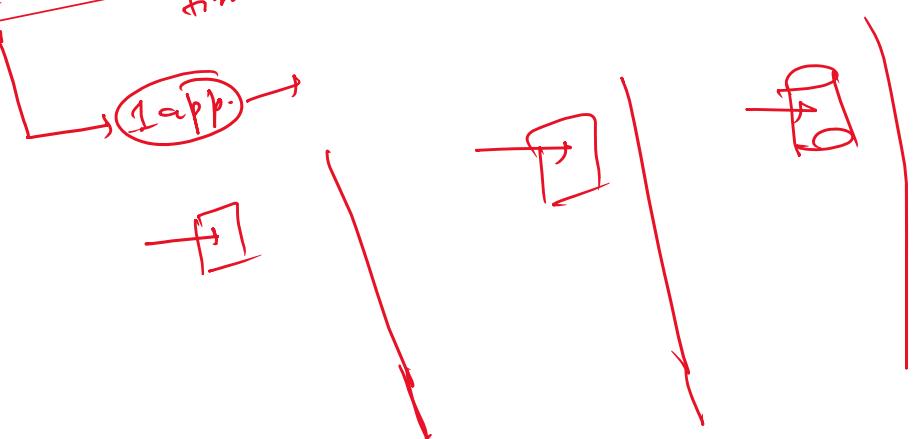


3 monitor Mode! -

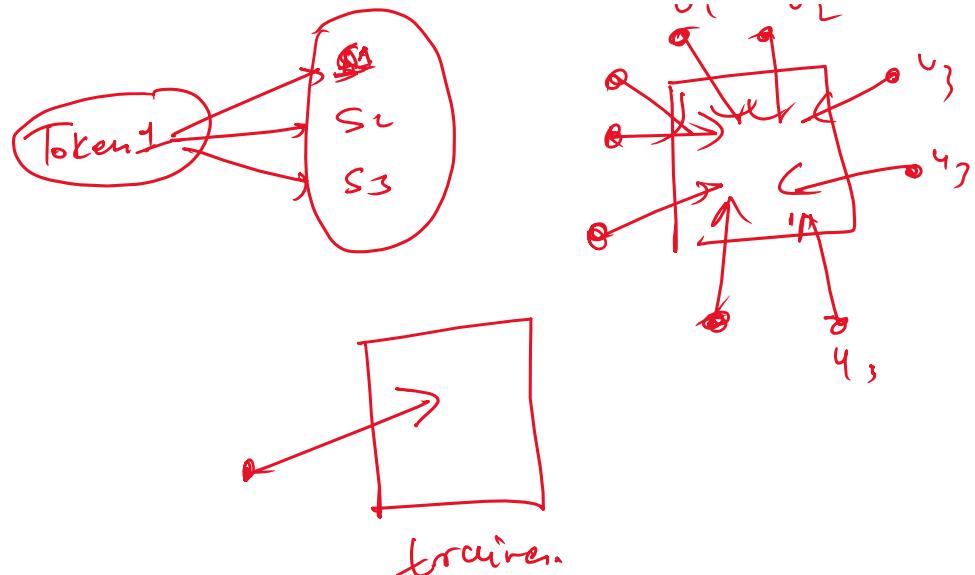
- ① Infra monitoring → Infra level metrics
- ② fullstack monitoring → Monitor the end-to-end App - along with RUM (real user monitoring)
- ③ Discovery mode → High level metrics Info.

Token:-

- = ① on the go - every time, it will create the fresh token.
- = ② Access Token - we can reuse the token multiple times.

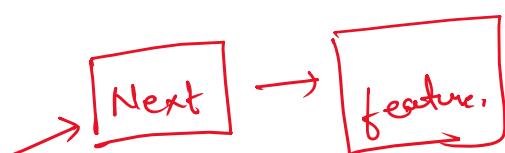


$$U_1 \quad U_L \\ \alpha_1, \dots, \alpha_L \quad U_2$$



- ① Metrics - 14 days
- ② Distributed traces - 10 days
- ③ Service Aggregates - 35 days
5-30 days
- ④ Logs - 35d
- ⑤ RUM Session - 35d
- ⑥ Dashboard - 35d
- ⑦ Smartscope - 35d.

XHR - XML http Request
fetch API calls.



↑ full page reload.
only a feature will pop out.

- ① Measure response time of asyn call
- ② Detect error in API response.
 - ~ 1 action for . u_)

- ② Detect errors in front end
- ③ correlates fronted action to backend service (via prep path)

INP - Metric defined by Google.

- ① User centric performance metric of the web page to user interaction.
- ② Responsiveness



INP:

- $\leq 200ms$ - good
- $200-500ms$ - Need Improvement
- $> 500ms$ - poor

High INP = Sluggish / frustrating user experience

$$PL = 1.2s$$

$$RTT = 800ms$$

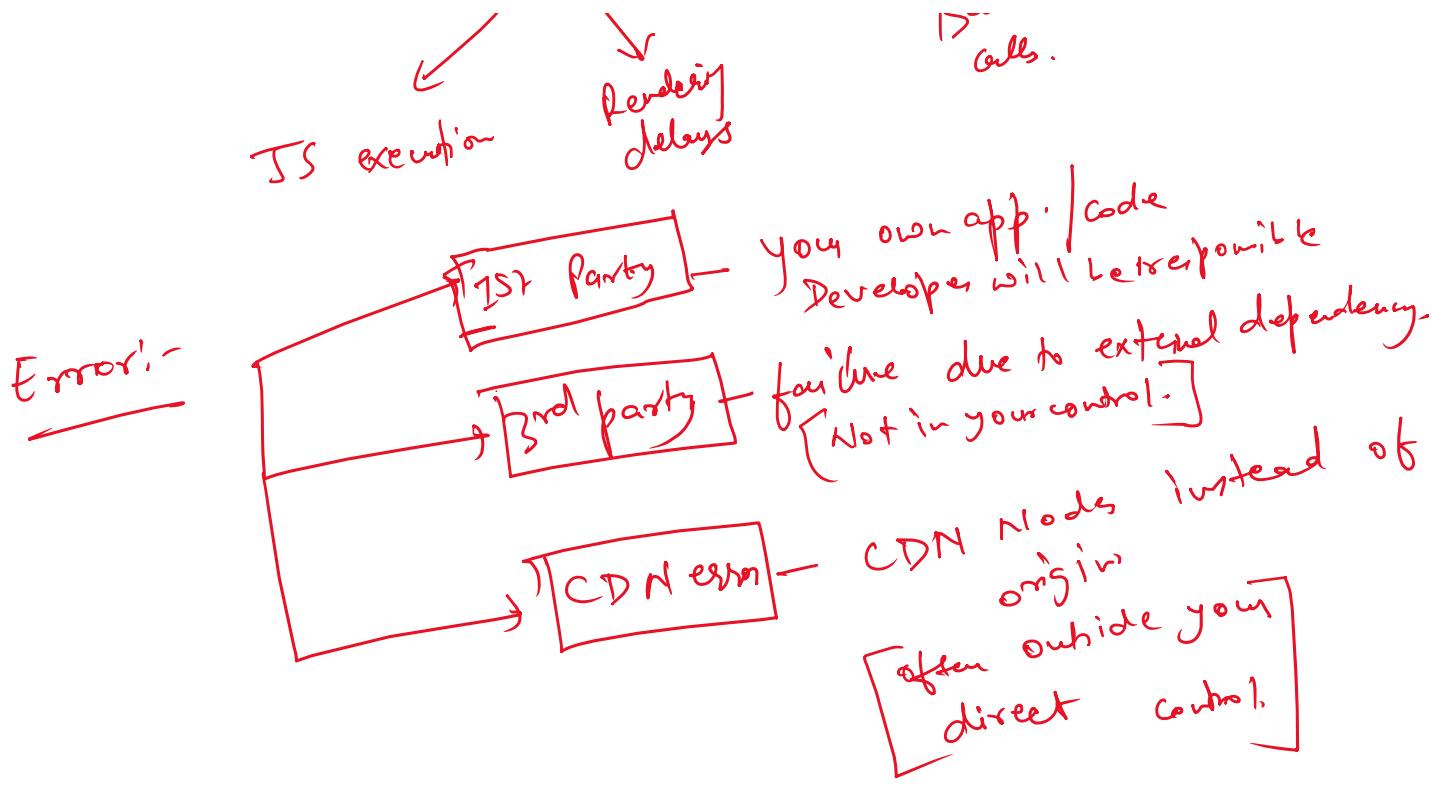
$$INP = \underline{650ms} \rightarrow \text{fetched } 5100\text{ ms}$$

How?

Breakdown INP contribution

Networking

Backend calls



CDN — Content delivery network.

Cache & deliver content (css, js; images, videos)

CDN error — Browser fails to fetch resources from CDN nodes instead of origin server.

