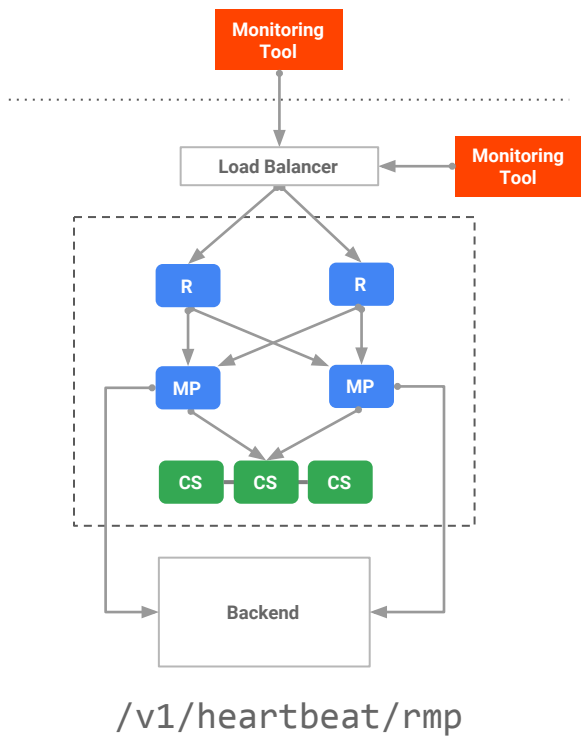


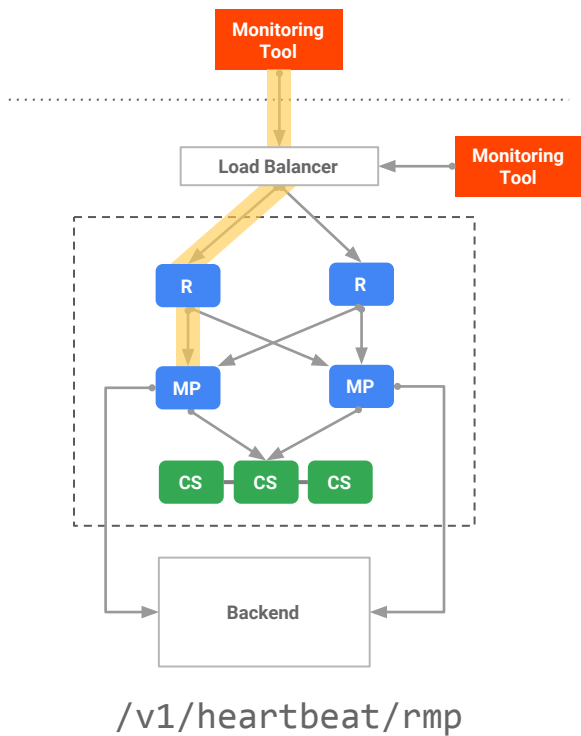


Synthetic Transactions Monitoring

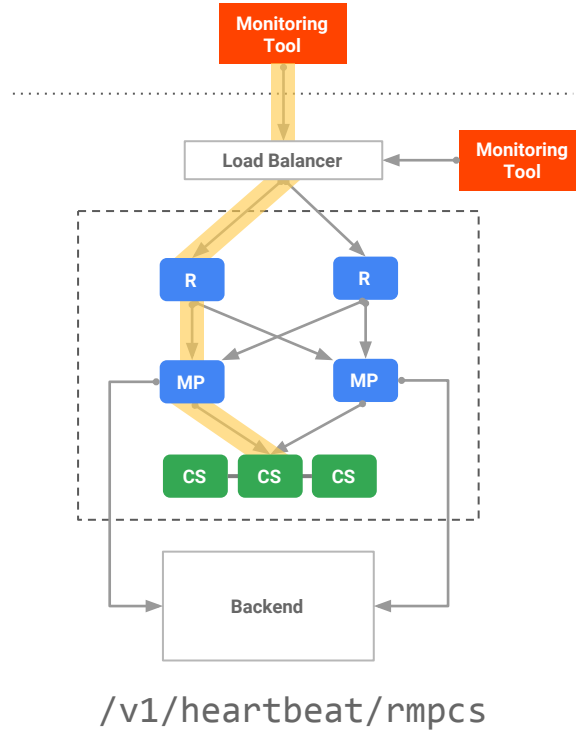
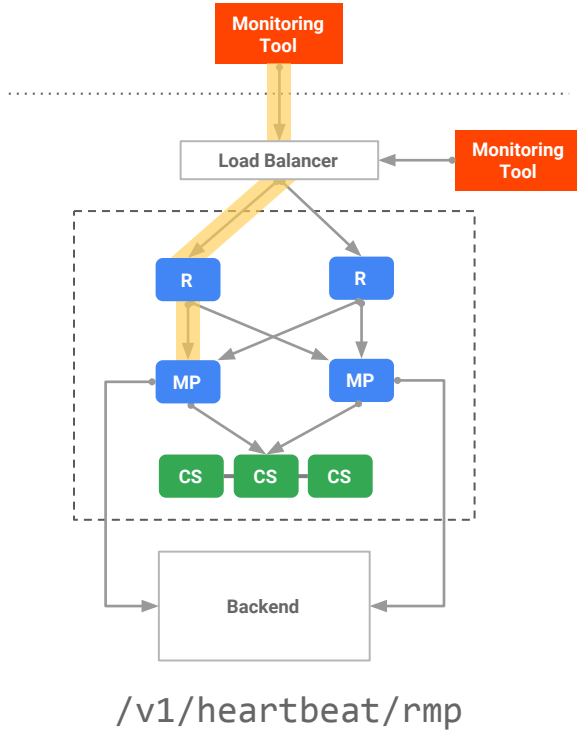
Synthetic monitoring



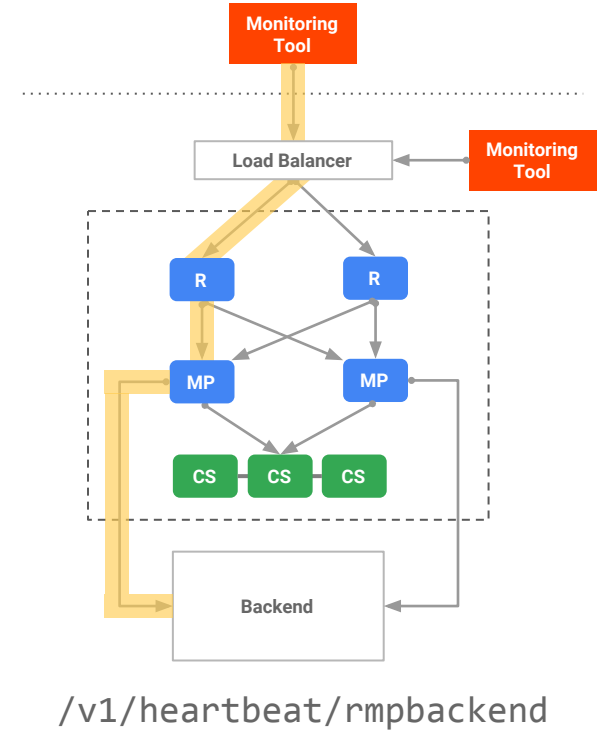
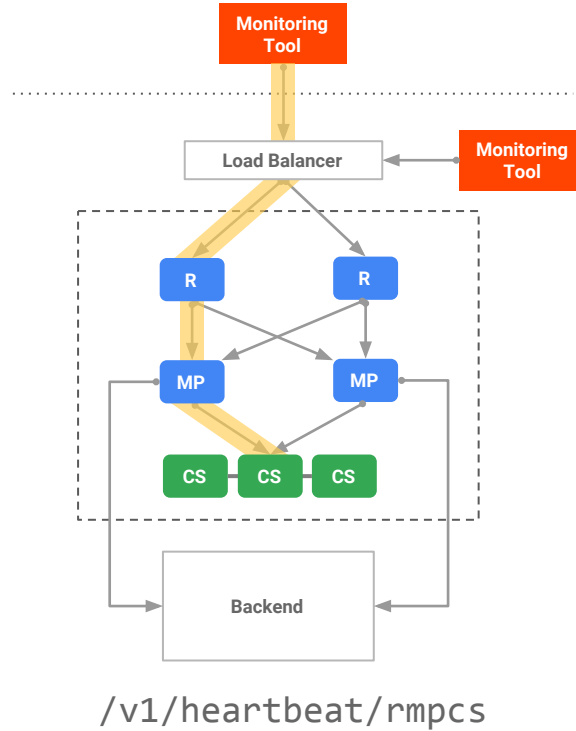
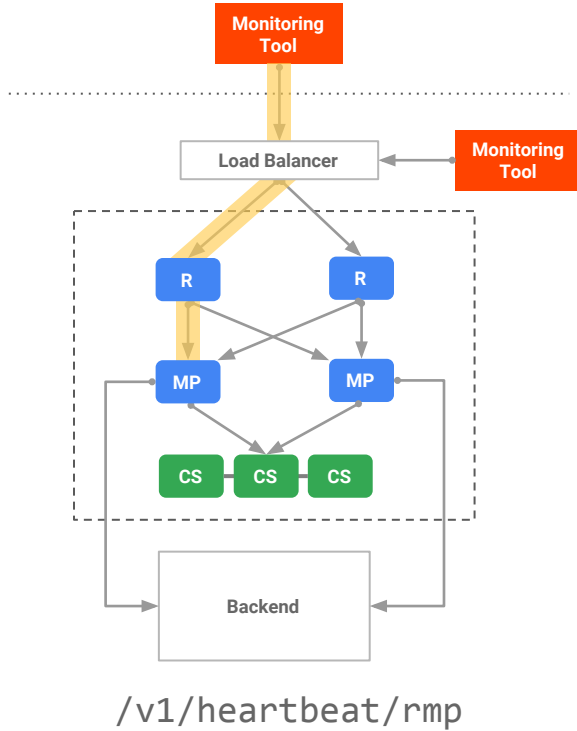
Synthetic monitoring



Synthetic monitoring



Synthetic monitoring



Heartbeat API proxy

Heartbeat API Proxy is distributed as part of the training material. The proxy is located on a directory called: 02a - Heartbeat API proxy.

1. Import heartbeat-api-proxy.zip to the organization.
2. Update heartbeat-api-proxy definition to match available virtual hosts on your environment.
3. Create a target server with the following name: HeartBeatTarget. The target server should point to any available HTTP resources in the backend system. Remember the goal here is not to test the backend systems or the backend systems APIs but Edge availability to connect to the backend system.
 - heartbeat-api-proxy implementation assumes the backend URL exposes by the target servers provides an operation called rmpbackend. Any call to /v1/heartbeat/rmpbackend proxy operation results on a call to <target>/rmpbackend.
 - You can modify heartbeat-api-proxy to change its behavior and customize it to your needs.
 - <https://community.apigee.com/questions/4303/how-do-i-transform-the-path-between-the-proxy-and.html>
4. Deploy heartbeat-api-proxy to your environment.

Heartbeat API proxy – Usage

Supported operations:

Prove Router and Message Processor:

```
curl -v http://<host>/v1/heartbeat/rmp
```

Expected success response: HTTP/1.1 200 OK

Prove Router, Message Processor and Cassandra:

```
curl -v http://<host>/v1/heartbeat/rmpcs
```

Expected success response: HTTP/1.1 200 OK

Prove Router, Message Processor and Backend target endpoint:

```
curl -v http://<host>/v1/heartbeat/rmpbackend
```

Expected success response: HTTP/1.1 200 OK



Thank You