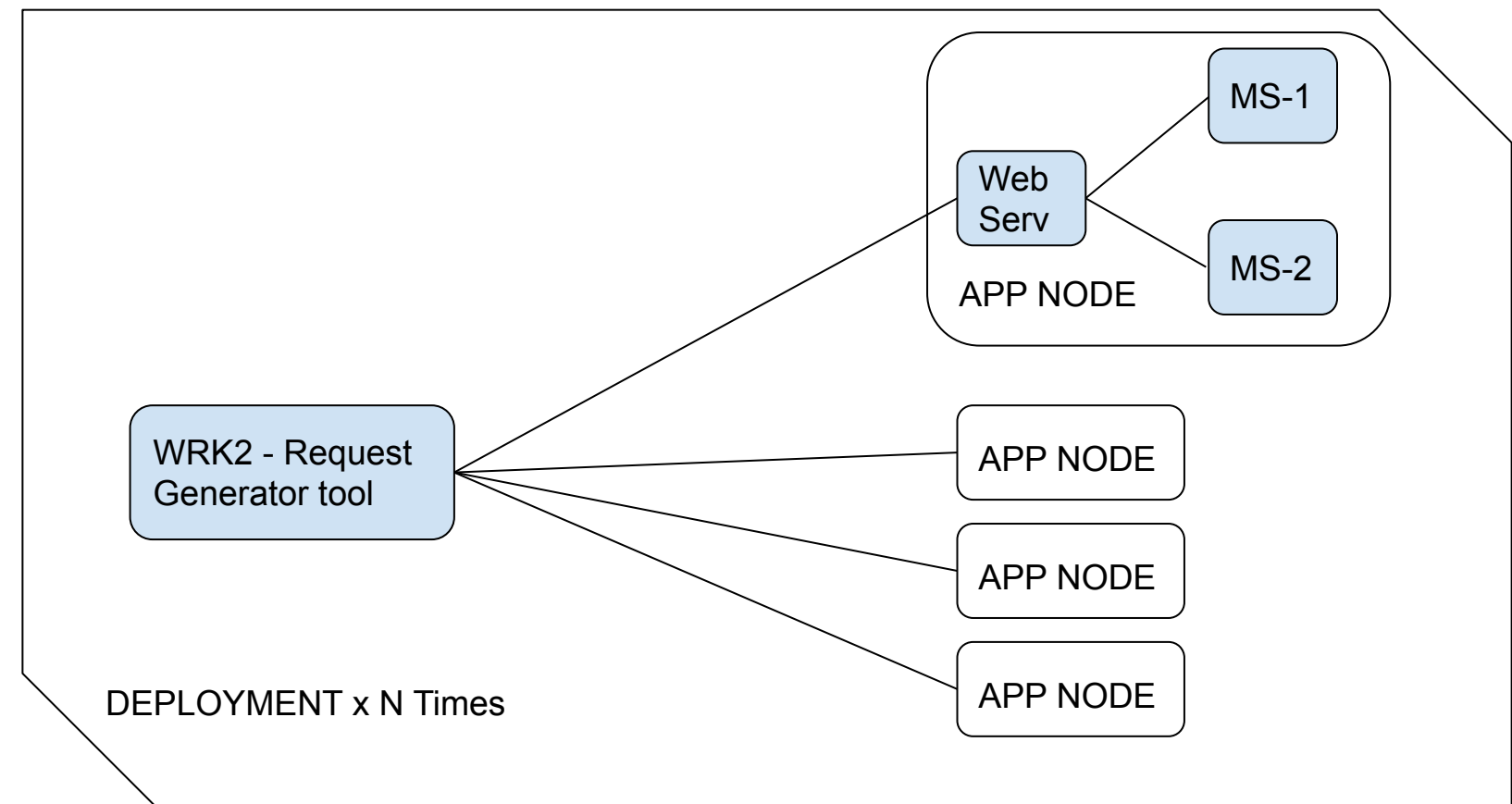


Kinvolk Study:
Each deployment contains 5 Nodes, their request generator tool in Node 1, sends simultaneous requests to the 4 App Nodes. Each Node contains 3 Services. I believe they take data from worst performing connection

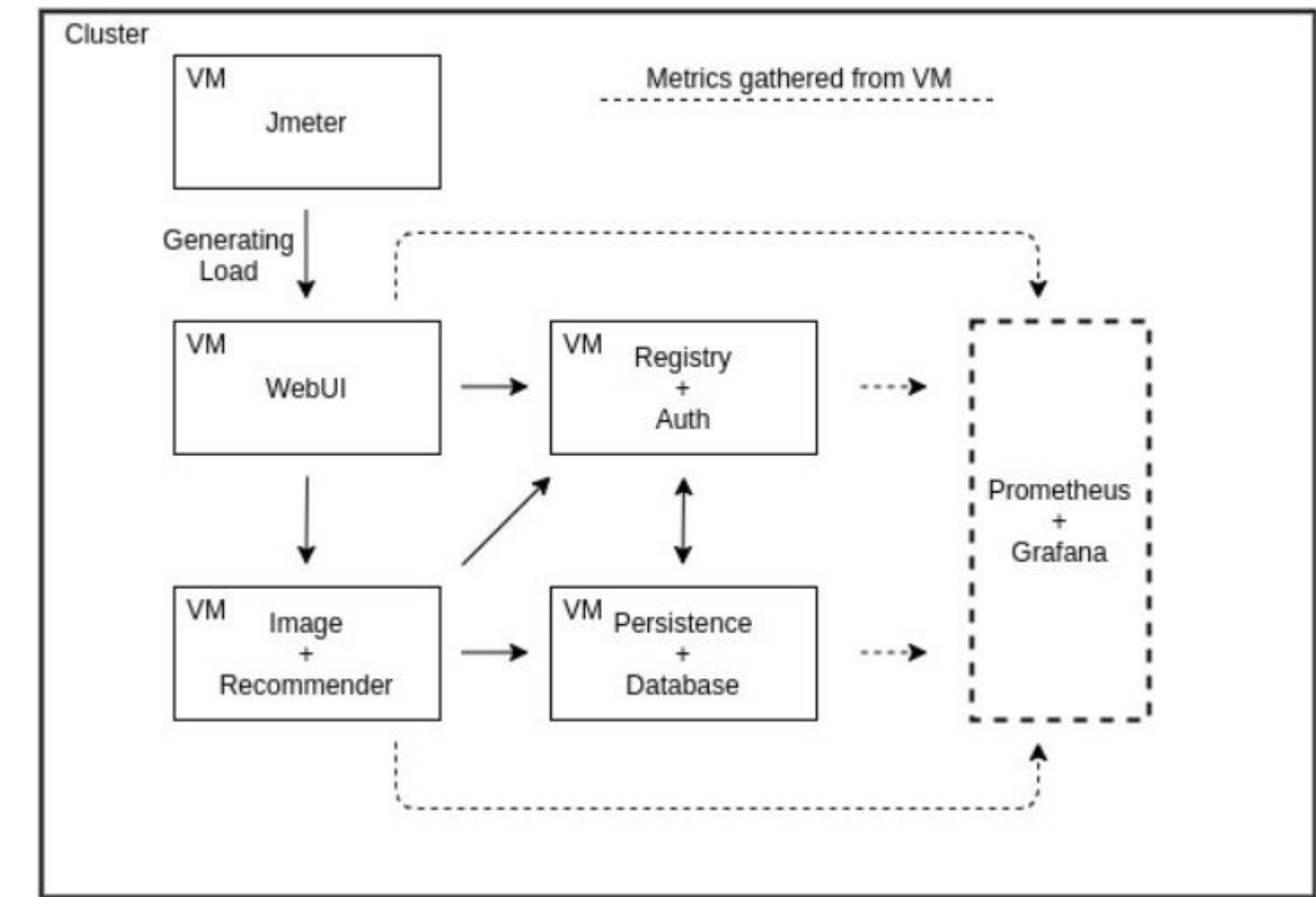
However each App Node only handles 1 request at a time. They do saturate the service by generating hundreds of requests, but only send a new one once they’ve heard back from the previous one.



Elastisys Study:
Erik used a more complicated architecture to represent a more realistic usage. Furthermore he made simultaneous request to the same application Node, therefore representing usage from multiple users.

By running as multiple users Erik has one App Node deal with more than 1 request at a time, making it more realistic.

Prometheus and Grafana are Dashboards used to view consumption of each pod (Not sure if we should count them as MS or not). However, per request done by JMeter we are only waiting on 4 micro-services to communicate.



CloudCover Study:
Builds on Kinvolk, more detailed data and includes Consul. I believe it uses the same architecture though, so per request we wait on 3 services

My Idea:
Allow user to deploy as many MicroServices as they want, have them loop around a counter

E.G User wants to run X Microservices, have counter reach Y, and have the Network be used by Z users at a time.

This way we can test SMTs on multiple micro-services calls and not just be limited to 2-5. We can also stress test SMTs managing multiple networks by changing Z users.

