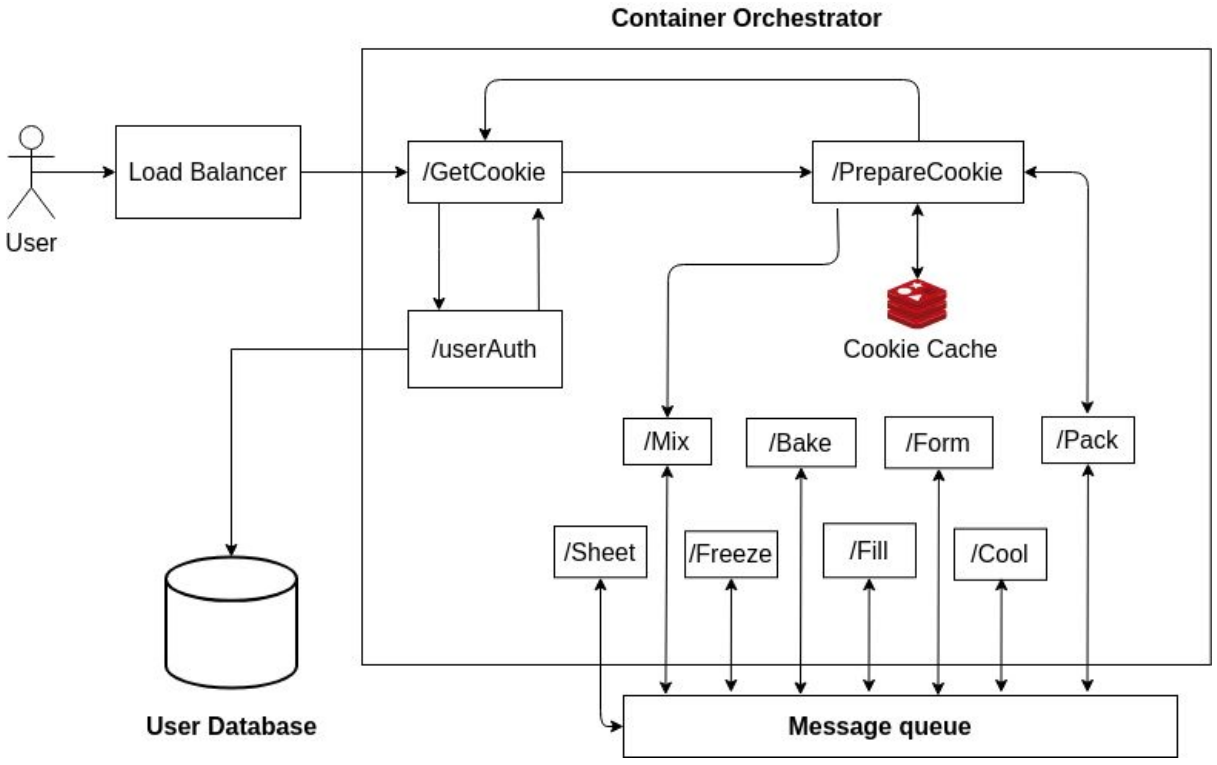
Intergalactic Cookies

Introduction

In this architecture one tries to be as cloud agnostic as possible. It's even possible to replicate this setup in an onprem environment if necessary. Please assume all components have proper redundancy (check section below).

Each rectangle with a "/verb" represents a microservice.



Redundancy:

All components have redundancy.

There are at least 2 Load Balancers. The microservices, if in containers, could be scaled up easily if Kubernetes is used (for example, by using horizontal pod autoscalers).

If the message queue chosen is Kafka then at least 3 nodes are necessary to reach quorum.

On what concerns databases, one can leverage replication of Amazon RDS. Otherwise, special configurations need to be put in place (eg.: master-master, master-slave, etc).

Intergalactic Cookies

Examples of technologies that can be used

Cookie Cache: Redis, Amazon DynamoDB
Container Orchestrator: Kubernetes, Nomad
Load Balancer: Amazon ELB, Azure ALB or F5
Message Queue: Kafka, Amazon SQS, RabbitMQ
Microservices: JavaScript, Go, .NET
User Database: Amazon RDS (MySQL, Aurora, other). MongoDB

Flow:

1. User requests a cookie by issuing a GET to /GetCookie.
2. /GetCookie microservice checks if user is allowed to get a cookie by consulting DB
3. If the above step returns 200, /GetCookie calls /PrepareCookie.
4. /PrepareCookie checks if there are cookies in the cache. If so, returns one from the cache, otherwise it triggers the /Mix microservice.
5. /Mix will start the work and send the result to the message queue.
6. All other microservices will pick-up work from the message queue as the results will likely come in an asynchronous way.
7. /Pack will finish up the job and send it back to /PrepareCookie which will return to /GetCookie and then to the user

Observations:

For inter-microservice communication outside of the message queue, we could use Istio which would provide encryption (mTLS) and tracing features among other security features.

All the infrastructure would be written as a code (ex.: Terraform). This would provide an easy way to migrate between cloud providers.