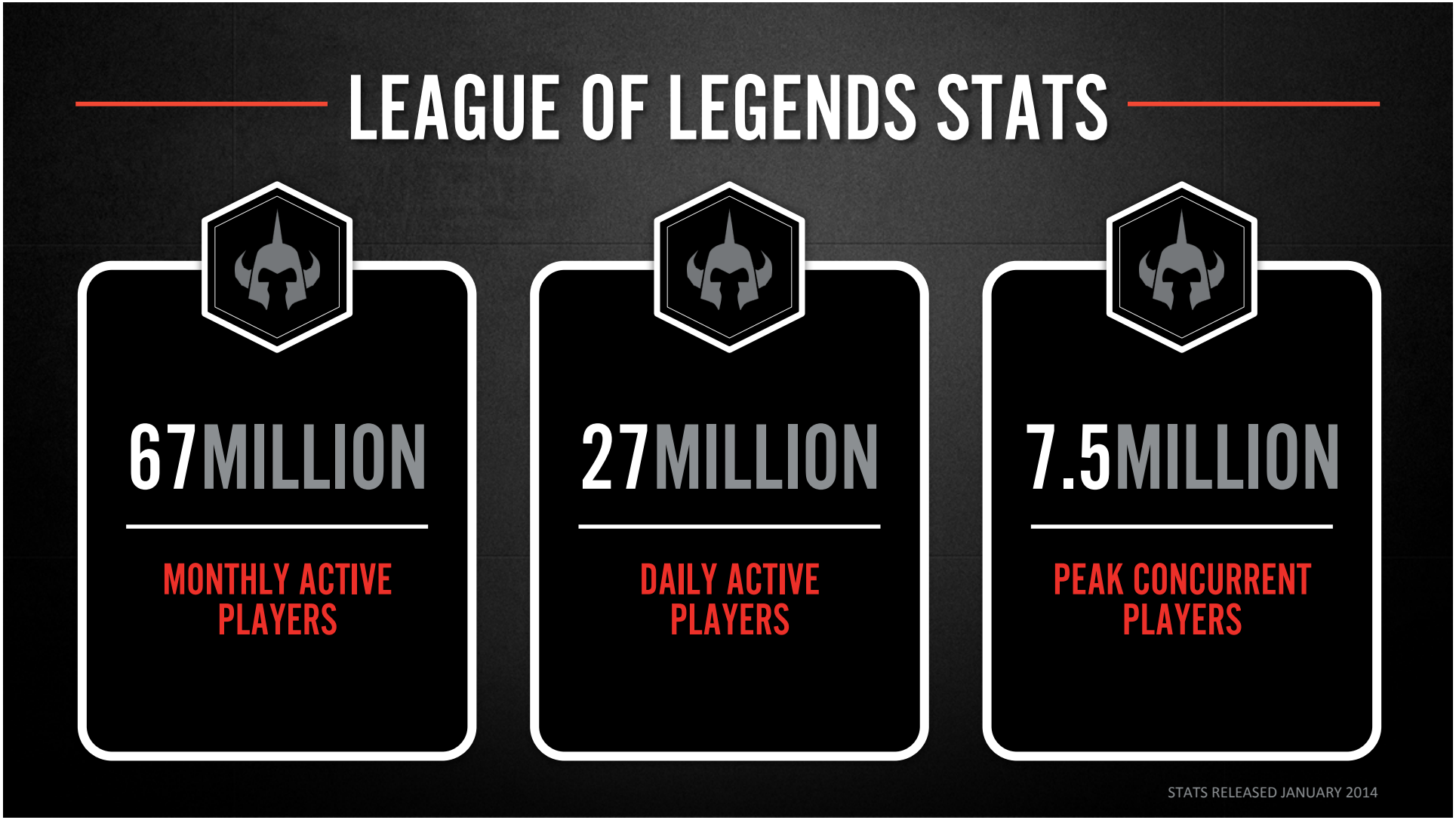
EXTENSIBLE REST & RPC COMMUNICATIONS FOR GAMES FEATURES

ANDREW MCVEIGH


LEAGUE OF LEGENDS


LEAGUE OF LEGENDS STATS
67MILLION
MONTHLY ACTIVE PLAYERS
27MILLION
DAILY ACTIVE PLAYERS
7.5MILLION
PEAK CONCURRENT PLAYERS
STATS RELEASED JANUARY 2014

OUR MISSION

WE ASPIRE TO BE THE MOST PLAYER FOCUSED GAME COMPANY IN THE WORLD

TODAY'S ROADMAP

THE PROBLEM

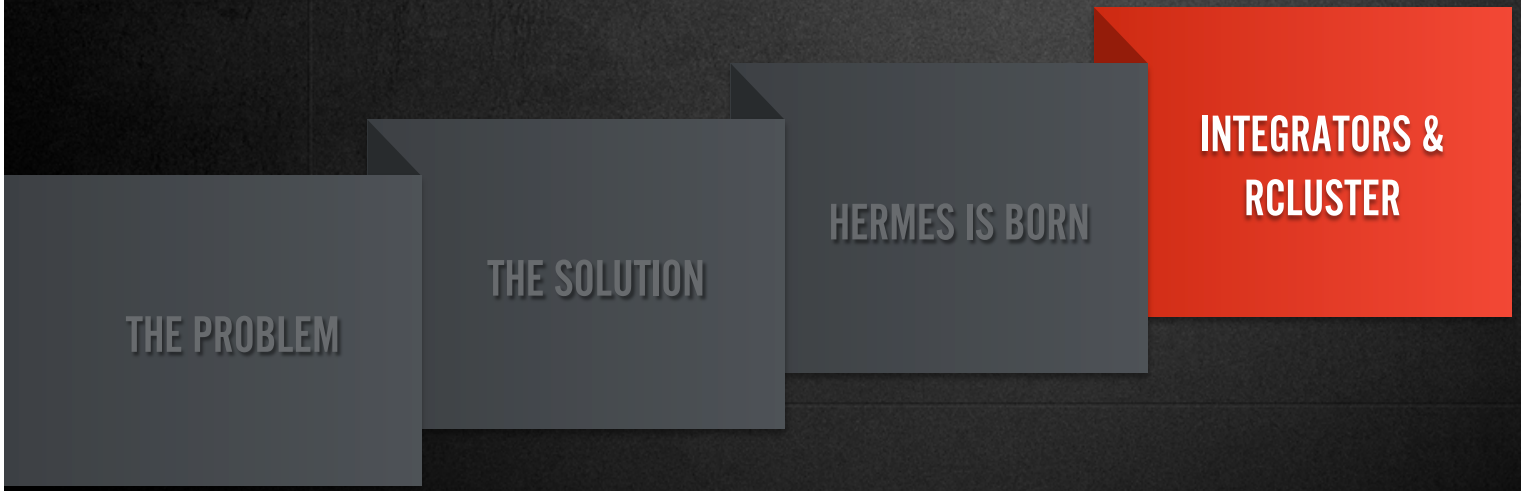
TODAY'S ROADMAP

THE PROBLEM

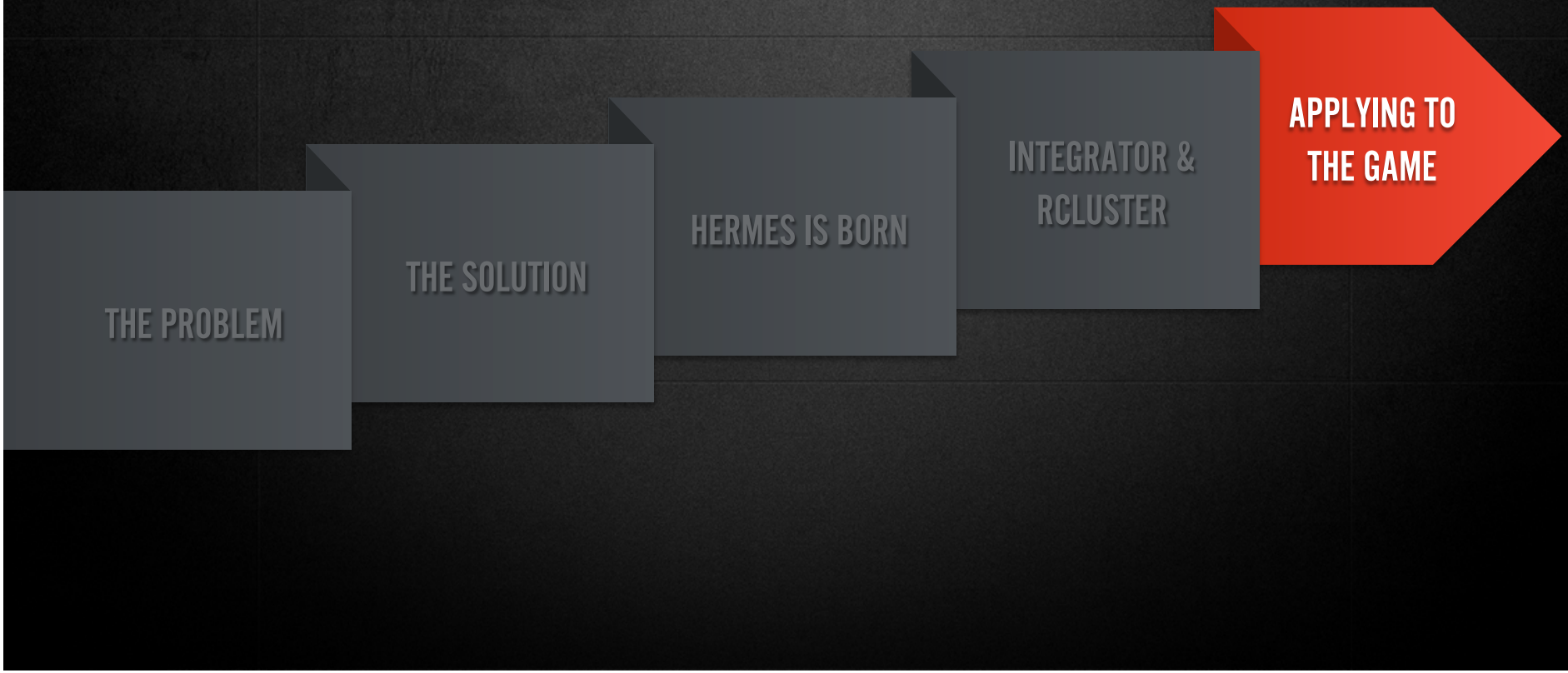
THE SOLUTION

TODAY'S ROADMAP

TODAY'S ROADMAP



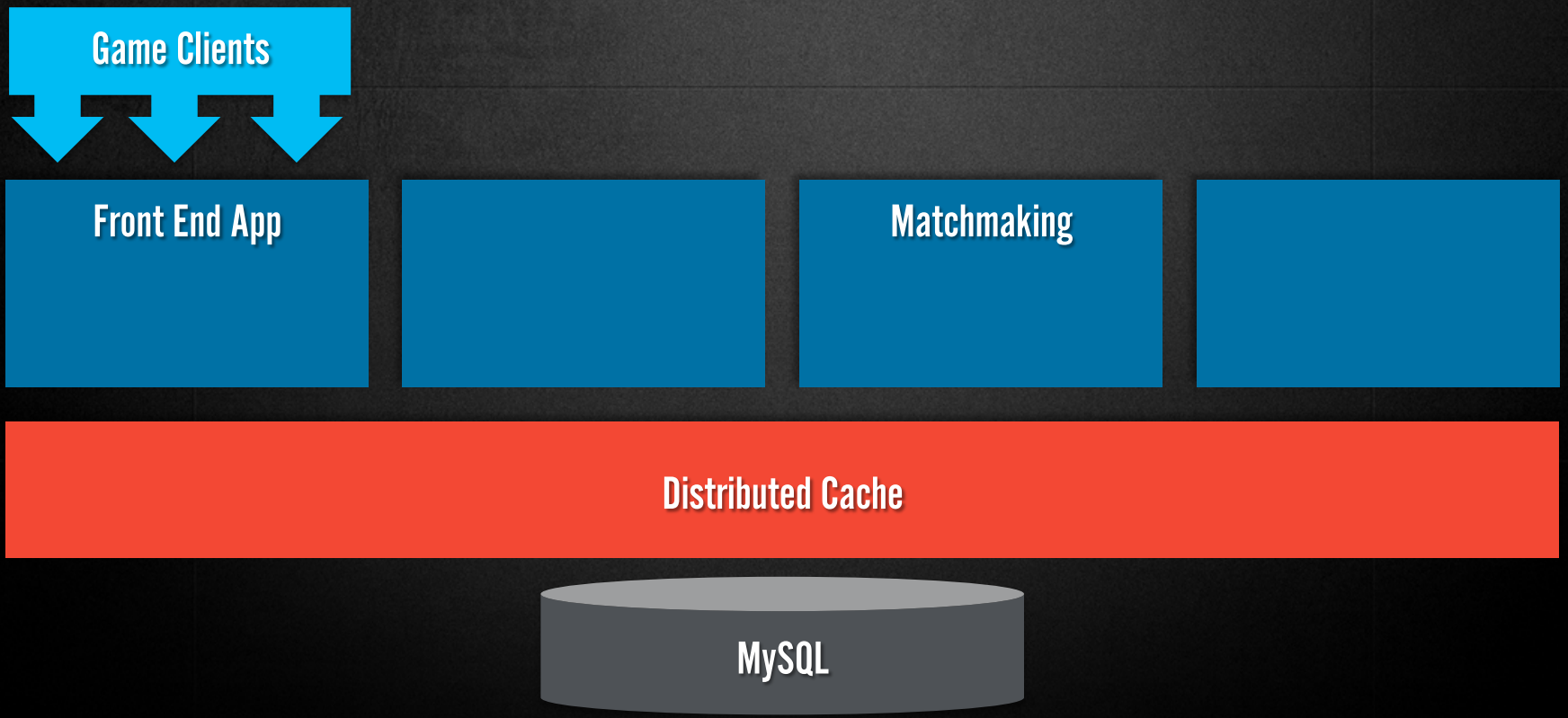
TODAY'S ROADMAP



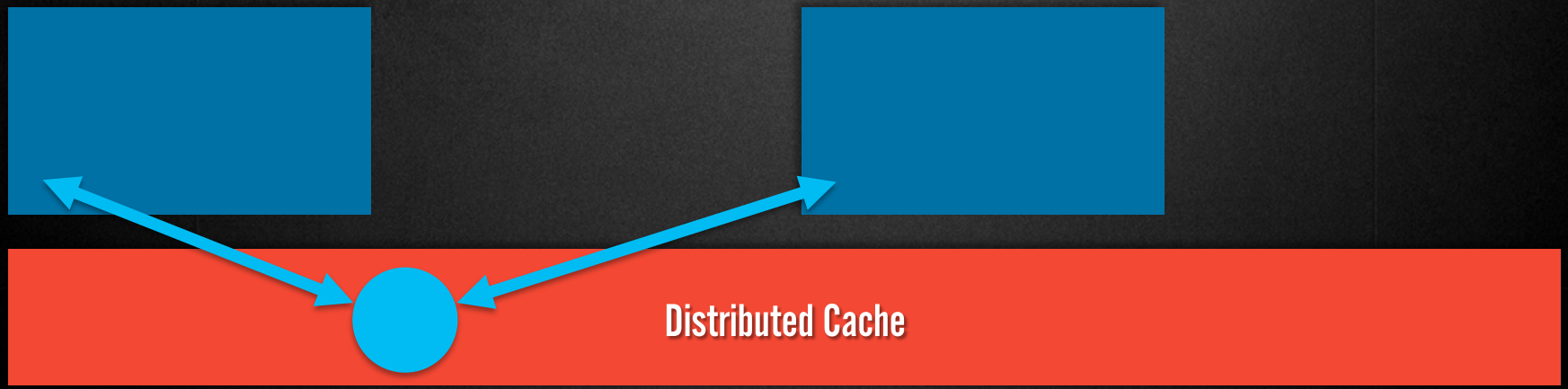
THE PROBLEM

INITIAL ARCHITECTURE

THE PROBLEM



RPC VIA CACHING



EXPLICIT COMMS

THE MOVE TO MICRO SERVICES

Service communications became very important

- Built on Twitter's Finagle initially
- Problems with interop, weight, and complexity
- Couldn't agree on a common approach
- Limited the infrastructure we could build...

THE SOLUTION

SERVICE COMMUNICATIONS RFC

Evaluated options to find a winner

- Created dozens of criteria and weighted them
- Big concern was performance and latency
- Scored Mercury, Thrift, JSON-RPC, REST/HTTP, others

AMBASSADOR FTW

AMBASSADOR

Spec first... then implementations

- HTTP + JSON + date formatting rules
- Covers REST and RPC
- Implement any way you want!

AMBASSADOR FTW

Separate spec & implementations

- HTTP + JSON + interop rules
- Covers REST & RPC
- Implement any way you want!

SWAGGER TO CREATE "LIVING APIS"

AMBASSADOR FTW

/purchase-v1-entitlements

Show/Hide | List Operations | Expand Operations | Raw

GET /{region}/v1/entitlements/{accountid} — Get Entitlements (secured with scopes: riot.store.purchase.view) (REST)

Response Class

List[EntitlementDTO]

class EntitlementDTO(inventoryType: string = ['BOOST' or 'BUNDLES' or 'CHAMPION' or 'CHAMPION_SKIN' or 'RP' or 'RUNE' or 'SPELL_BOOK_PAGE' or 'SUMMONER_CUSTOMIZATION' or 'SUMMONER_ICON' or 'WARD_SKIN'], items: List[EntitlementItemDTO]**)**

Parameters

Parameter	Value	Description	Data Type
AUTHORIZATION		Auth header: (Mock: Mock mocktoken) (GasToken: GasToken encodedgastoken)	string
region	MAIN1	Player Region	string
accountid	(required)	**Platform Account Id (authPrincipalType: Account)**	long
product	LOL	Product requesting entitlements	string
inventorytype	BOOST	Type of inventory to retrieve	string

Try it out!

HERMES IS BORN...

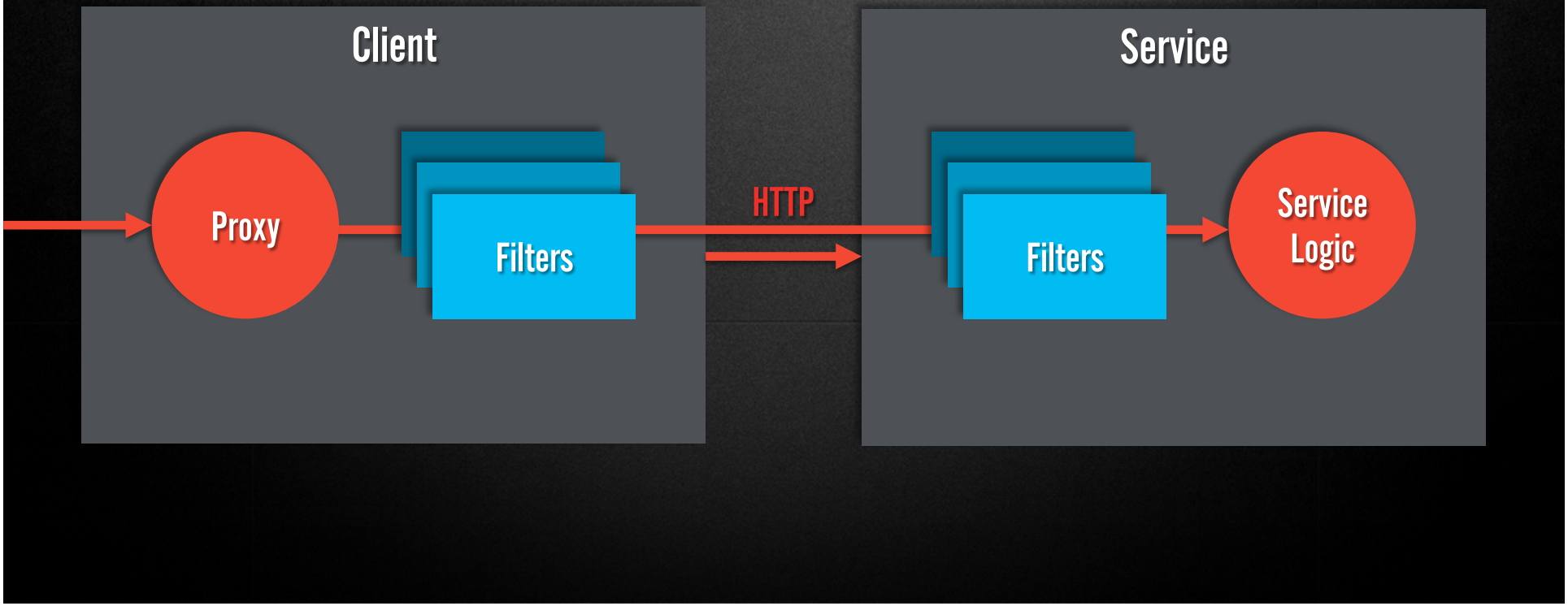
HERMES

Java implementation of Ambassador

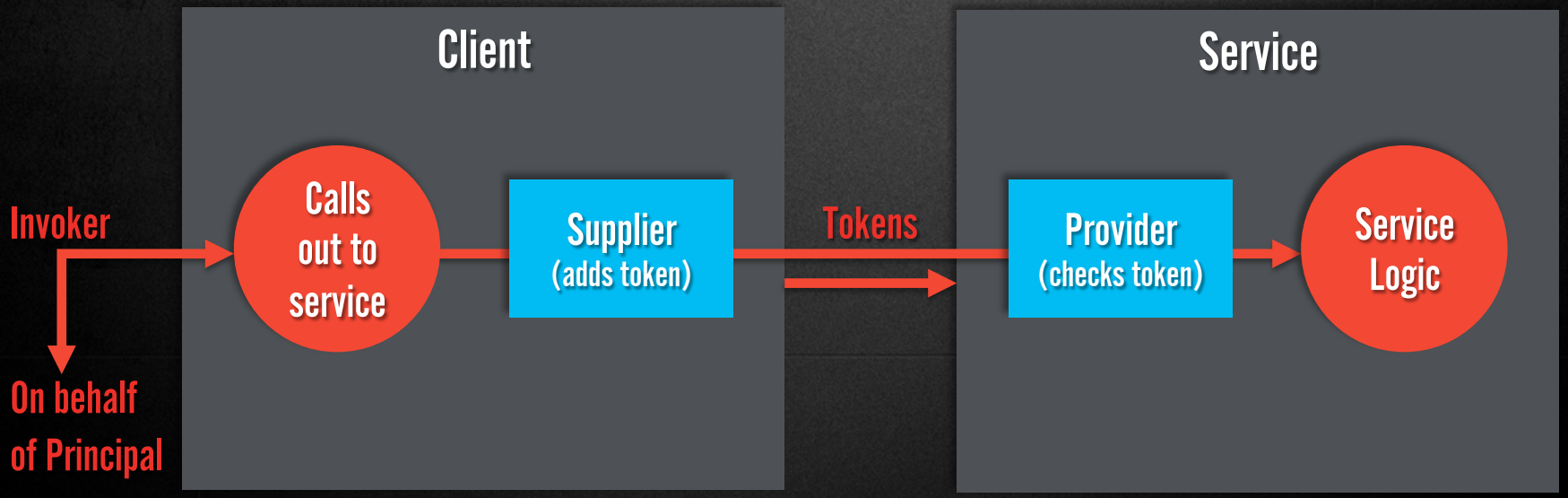
- Server side → Jersey + standard idioms
- Client side → Auto-generated proxy
- Spiked out in a week to get buy-in
- 22 Riot contributors

FILTERS FOR EXTENSIBILITY

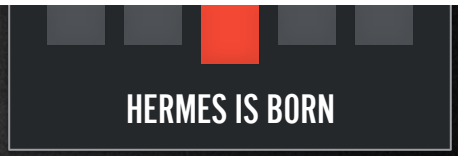
HERMES IS BORN



AUTH VIA FILTERS



START WITH THE CONTRACT

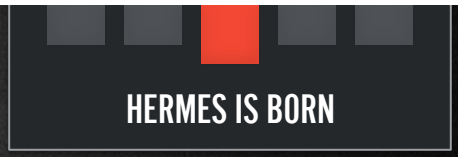



```
@HermesContract(
    approach=REST, name="group.contract", version="1.0.0",
    description="Tutorial example", errors = {"501|Problem logging"})

public interface TutorialContract {

    @POST
    @Path("v1/printMessage")
    @HermesOperation(summary="Print a message", version="1.1")

    void printMessage(@QueryParam("message") String message);
}
```

SERVICE CODE

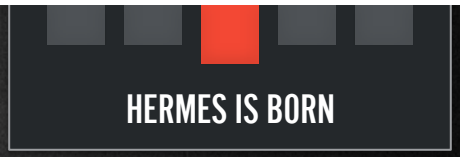

```
@Path("tutorial")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
@Slf4j
public class TutorialService implements TutorialContract {

    public void printMessage(String message) {
        log.info("Message: " + message);
    }
}
```

STARTING THE SERVER



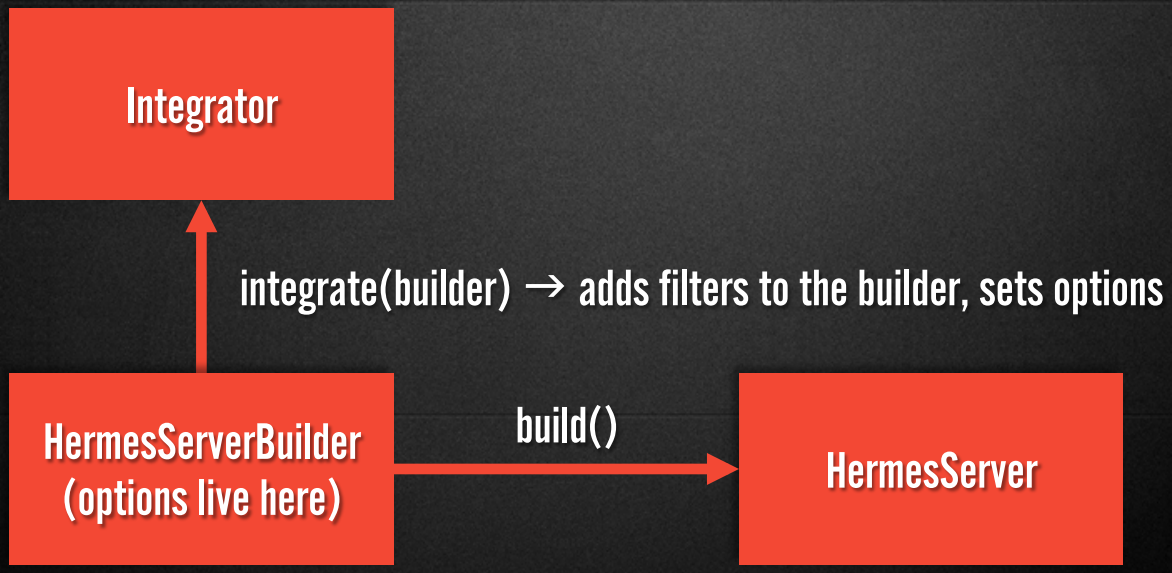

```
HermesServer server = HermesNettyServerBuilder.newBuilder()
        .singletons(new TutorialService())
        .base("tutorial")
        .requestFilters(...)
        .responseFilters(...)
        .integrators(...)
        .build();
```

CLIENT CODE



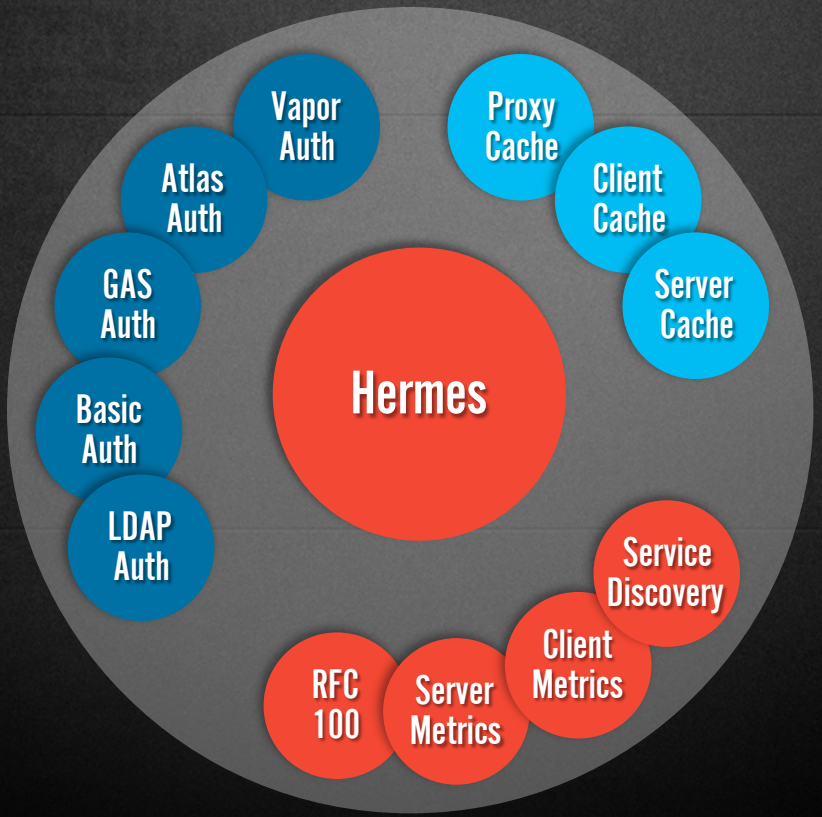

```
HermesClient client = HermesClientBuilder.newBuilder()
        .baseURIs("http://localhost:7772")
        .proxyContractClass(TutorialContract.class)
        .proxyPathHint("tutorial")
        .filters(new LoggingFilter())
        .integrators(...)
        .build();

TutorialContract proxy = client.createProxy();
proxy.printMessage("Hello world");
```

INTEGRATORS ALLOW ENCAPSULATION

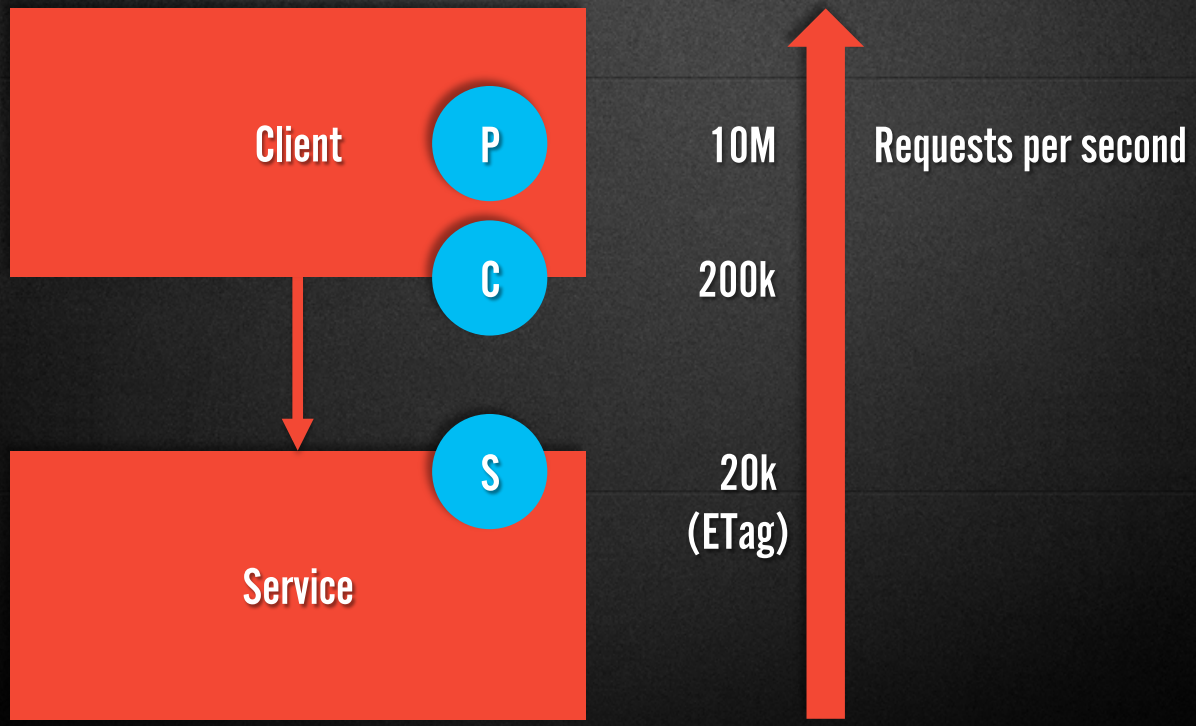


INTEGRATORS & RCLUSTER

PLUG & PLAY INTEGRATORS



CACHING INTEGRATORS

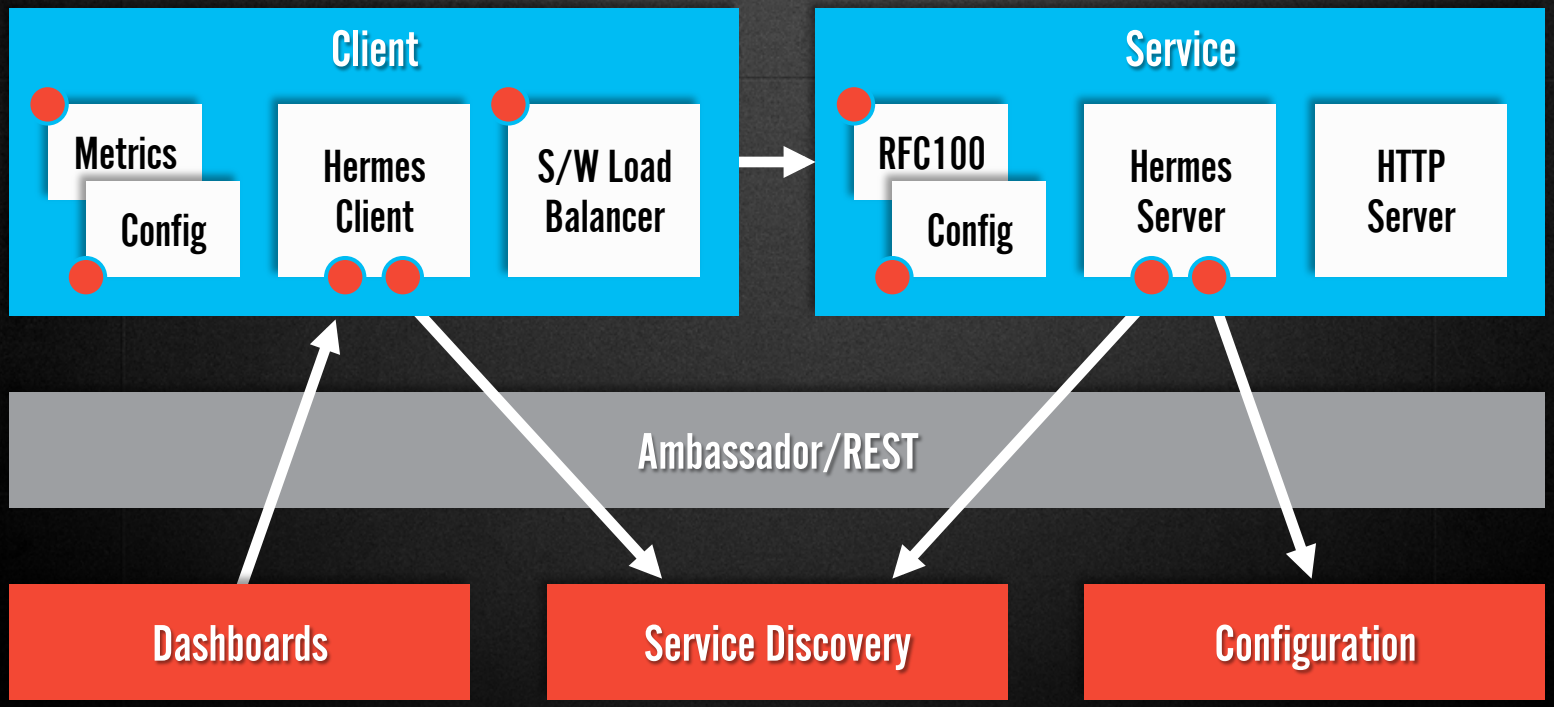


POLLING

Polling simplifies failure scenarios

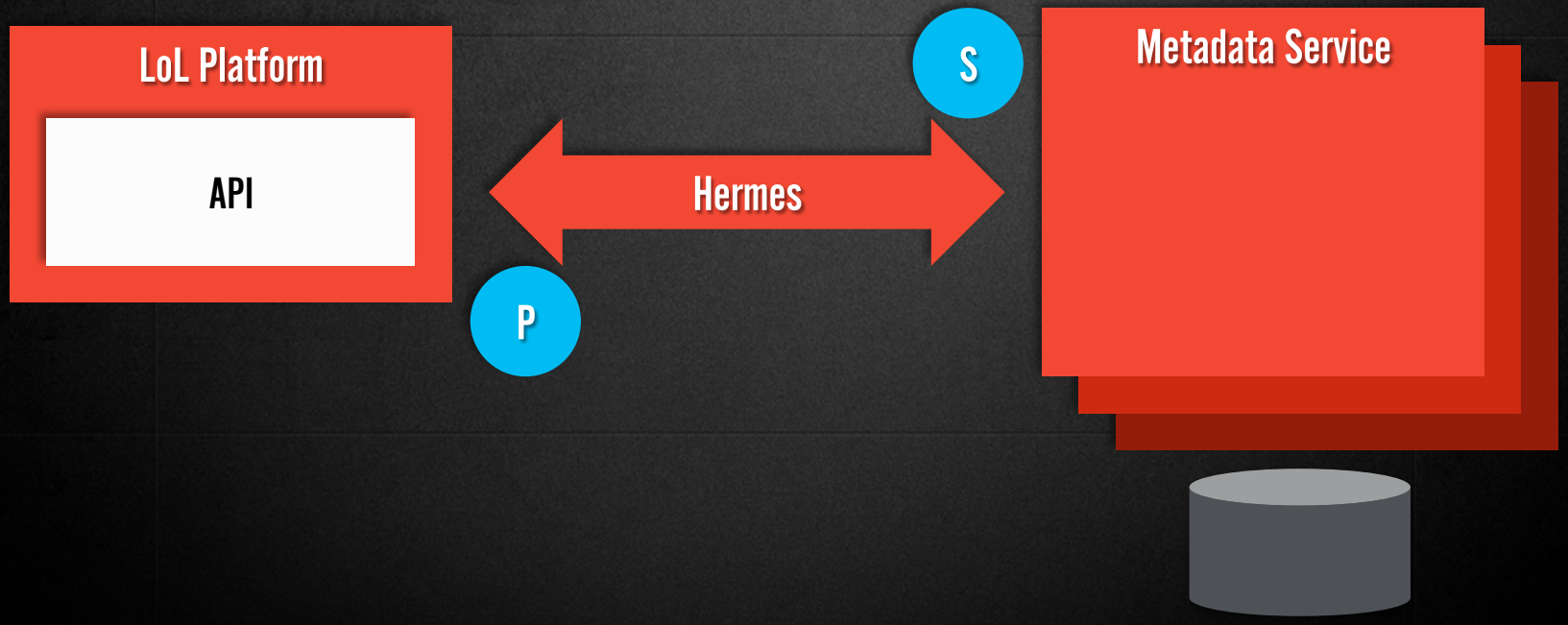
- Clients poll service
- Services don’t need eventing
- Services don’t remember connections!

RCLUSTER

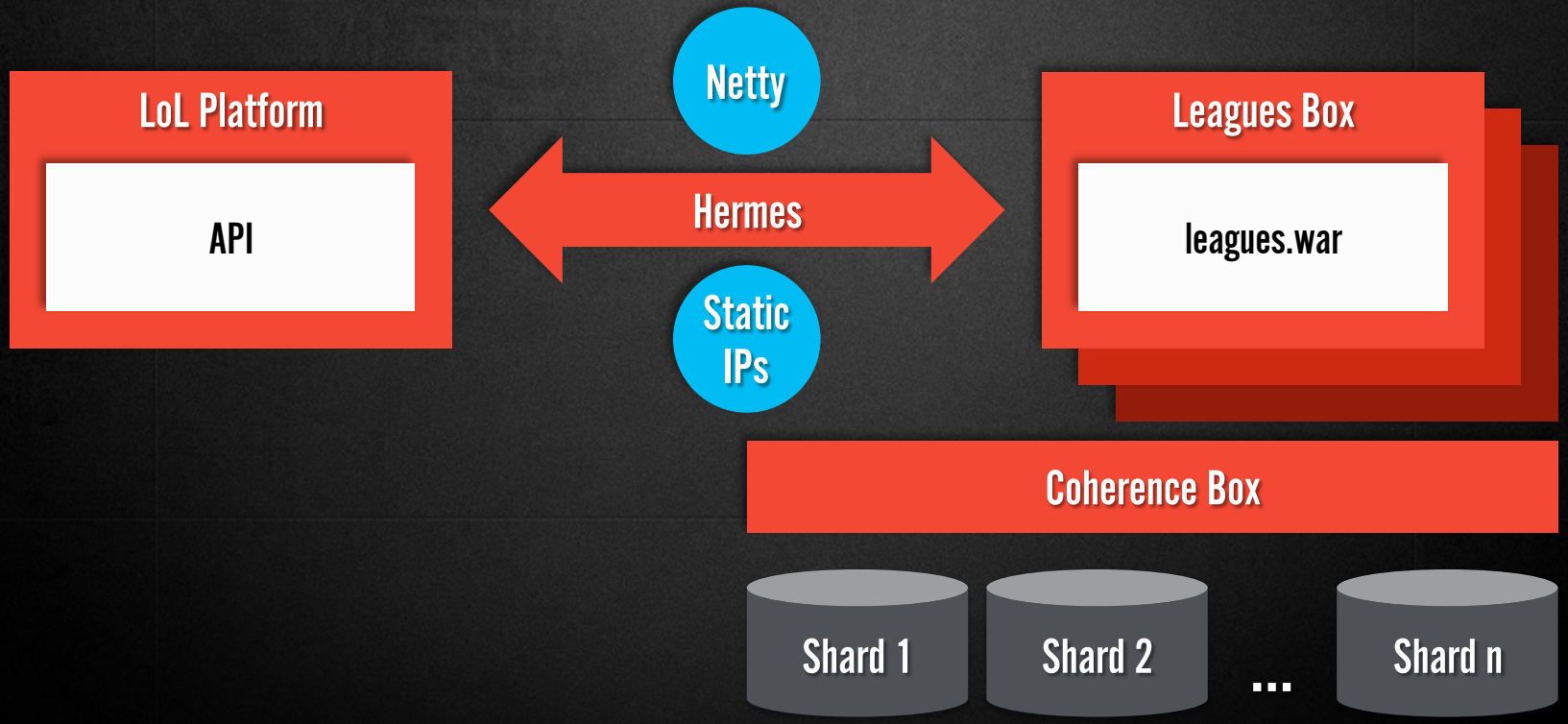

Client
Metrics
Config
Hermes Client
S/W Load Balancer
Service
RFC100
Config
Hermes Server
HTTP Server
Ambassador/REST
Dashboards
Service Discovery
Configuration

APPLYING TO THE GAME

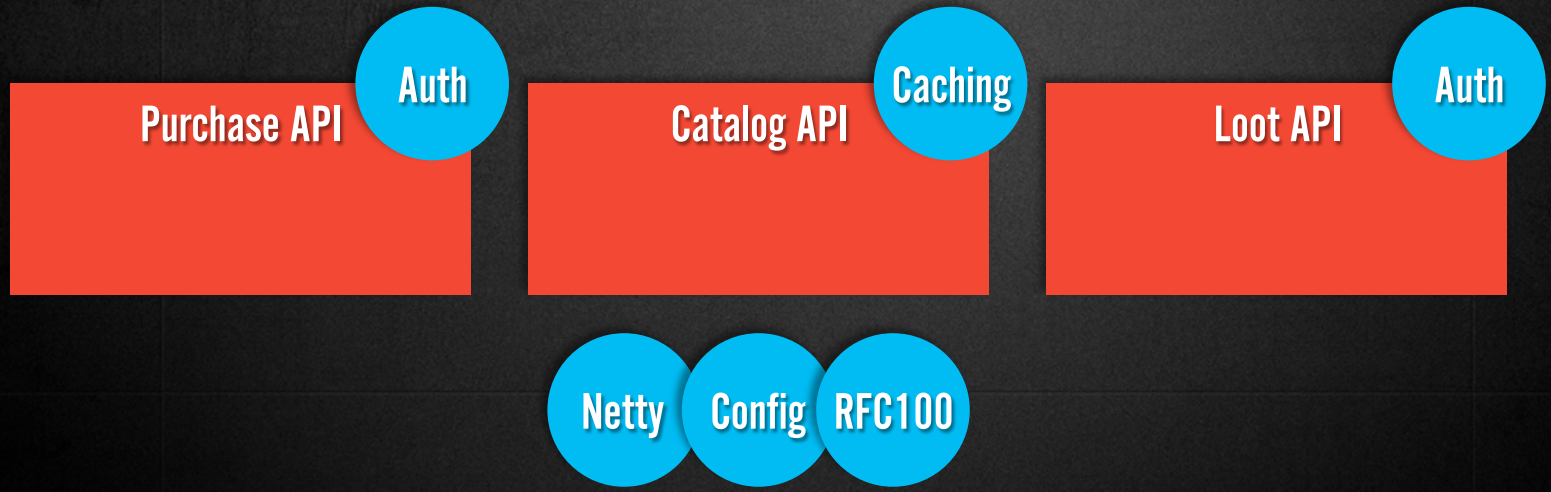
METADATA SERVICE



LEAGUES

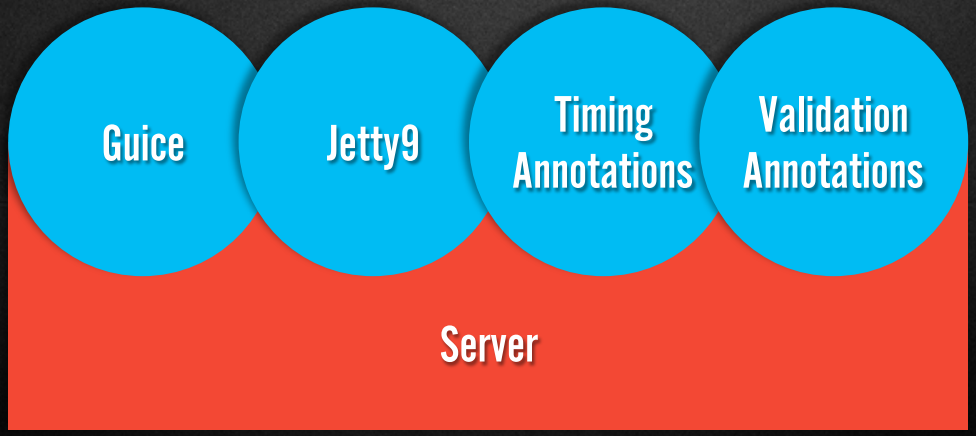

LoL Platform
API
Netty
Hermes
Static IPs
Leagues Box
leagues.war
Coherence Box
Shard 1
Shard 2
...
Shard n

STORE


Purchase API
Auth
Catalog API
Caching
Loot API
Auth
Netty
Config
RFC100

MOBILE TEAM

Heavily customized for their stack



FUTURE PROOFING

Hermes sucks!

FUTURE WORK

Async events + HTTP2

- Eventing for async communications
- RxJava + reactive patterns
- Jersey 2.x, HTTP2
- Open sourcing to build community of integrators

https://github.com/riotgames

QUESTIONS?

ANDREW MCVEIGH
Software Architect
Riot Games
amcveigh@riotgames.com