POWERED BY

UNREAL
TECHNOLOGY

Gears of War 3 Analytics:

Optimizing the Online Experience, or How I Learned to Stop Worrying and Love the Beta

Joe Graf (Dr. Statslove)

Game Connection 2011

December 8, 2011
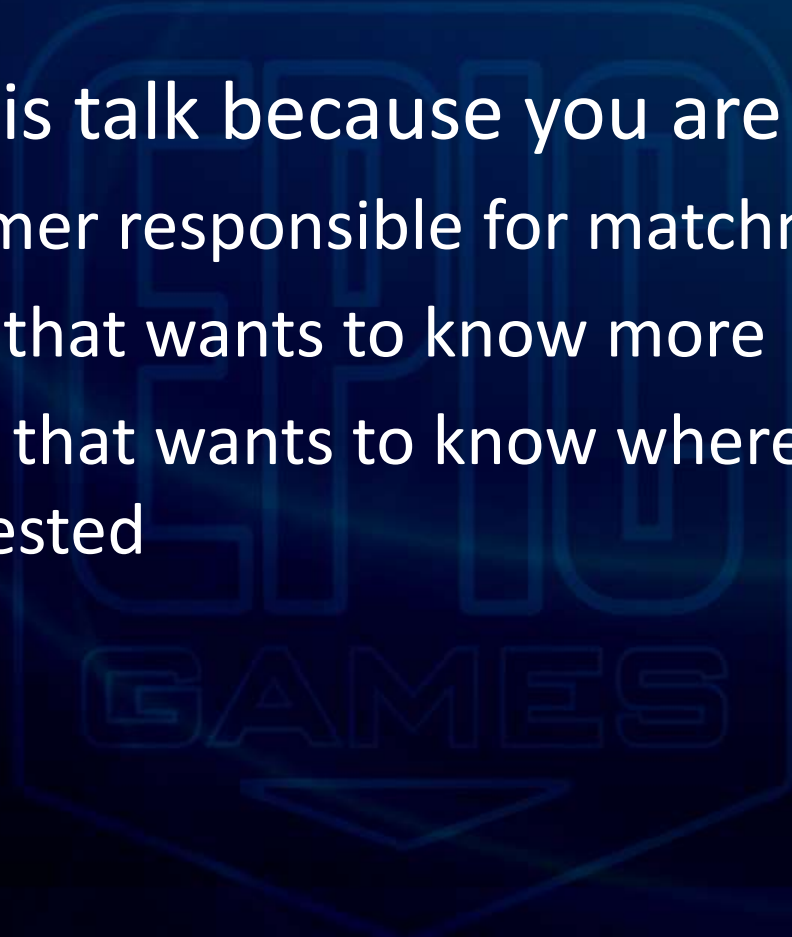
About Me

- Been at Epic over 8 years

- Primarily focused on online features for our games and engine
- Lead Gameplay Programmer for Gears of War 3

About You

You are at this talk because you are a

– Programmer responsible for matchmaking

– Designer that wants to know more

– Producer that wants to know where the Gears team invested

About Gears of War

- First two games were very popular multiplayer games

- Gears of War 2
  - 6 million units sold
  - More than 1 million simultaneous players

- Gears of War 3 looked to expand upon that via the inclusion of dedicated servers

Overview

- Network infrastructure of Gears

- Types of metrics the Gears beta captured

- How that data was used to pinpoint issues

- How the Gears matchmaking system works

- What we adjusted post beta for a great launch

Why do a beta?

- Gears 2 launch wasn't seamless
  - Overburdened Xbox Live services caused slow matchmaking at launch
  - Took about 1 month post ship to get everything solid

- Unanswered questions
 - Tested dedicated server support in Gears 2, but not at launch levels of players

Why worry about a beta?

- Risk!!!
  - Betas are seen as "Demos" not work in progress
    - Do it too early and pre-orders are cancelled
  - Schedule impact
    - Essentially shipping the game twice
    - Reduced output from team while beta is running
  - Throwaway work is polish lost on final product
    - Remove content not included in the beta
    - Custom UI flow and exposure to different TCRs

Goals for the Beta

- Find any scalability issues in matchmaking
  - Gears 2 did not exhibit issues with < 150k simultaneous players
  - Gears 2 test validated approach but not scalability
- Determine how many datacenters we need and where to locate them
- Test our data collection pipeline beginning to end
- Verify gameplay balancing with real players
- Test alternative matchmaking algorithms

Network Infrastructure
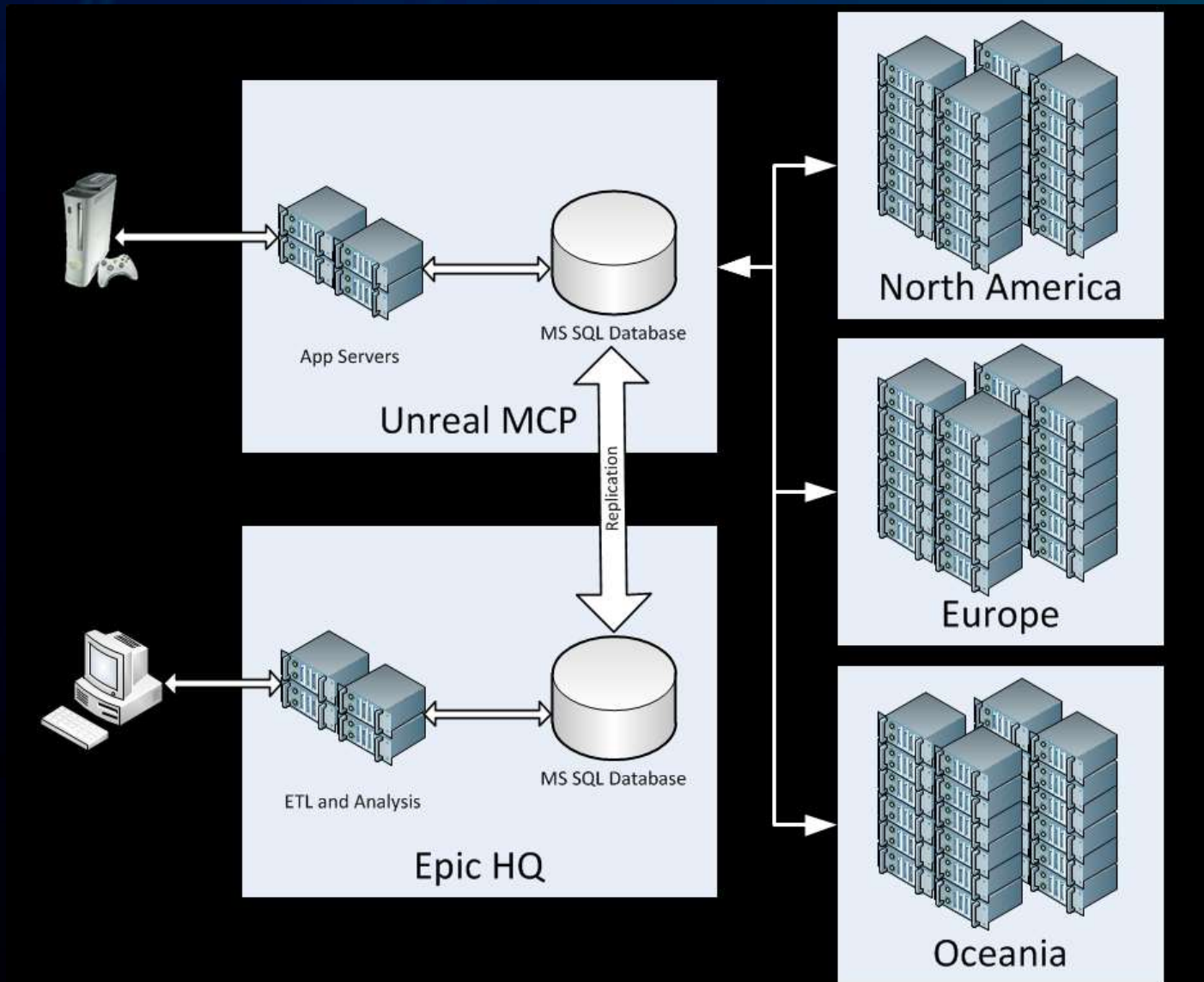
Unreal MCP

- Epic's Xbox Live Server Platform (XLSP)
  - Determines best datacenter for each player
  - Handles the playlist population tracking
  - Allows the Gears team to change data on the Xbox client without requiring a patch
  - Responsible for processing all uploads from the game

Data Captured

- Player data
  - Each player uploads their data as XML once per day
- Matchmaking data
  - Party host uploads an XML payload with the details
- Gameplay data
  - Each host uploads a compressed binary file containing all of the events that were not filtered out
- 1 Terrabyte of data collected in first week of beta

Player Data

- Population counts and location data
- Aggregate view of user preferences
  - MP character
  - Starting weapons and weapon skins
  - Button and stick configuration
    - Inverse, southpaw, legacy, etc.
- Player statistics and progress
  - Nearly 2,000 different data points

Player Data

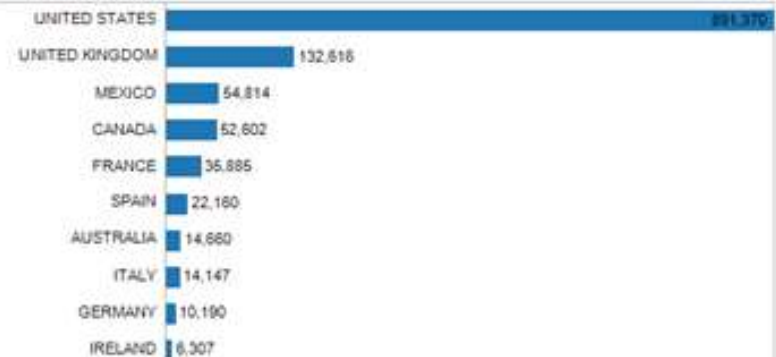Users by Country (Last Update: 5/25/2011 10:49:04 AM)

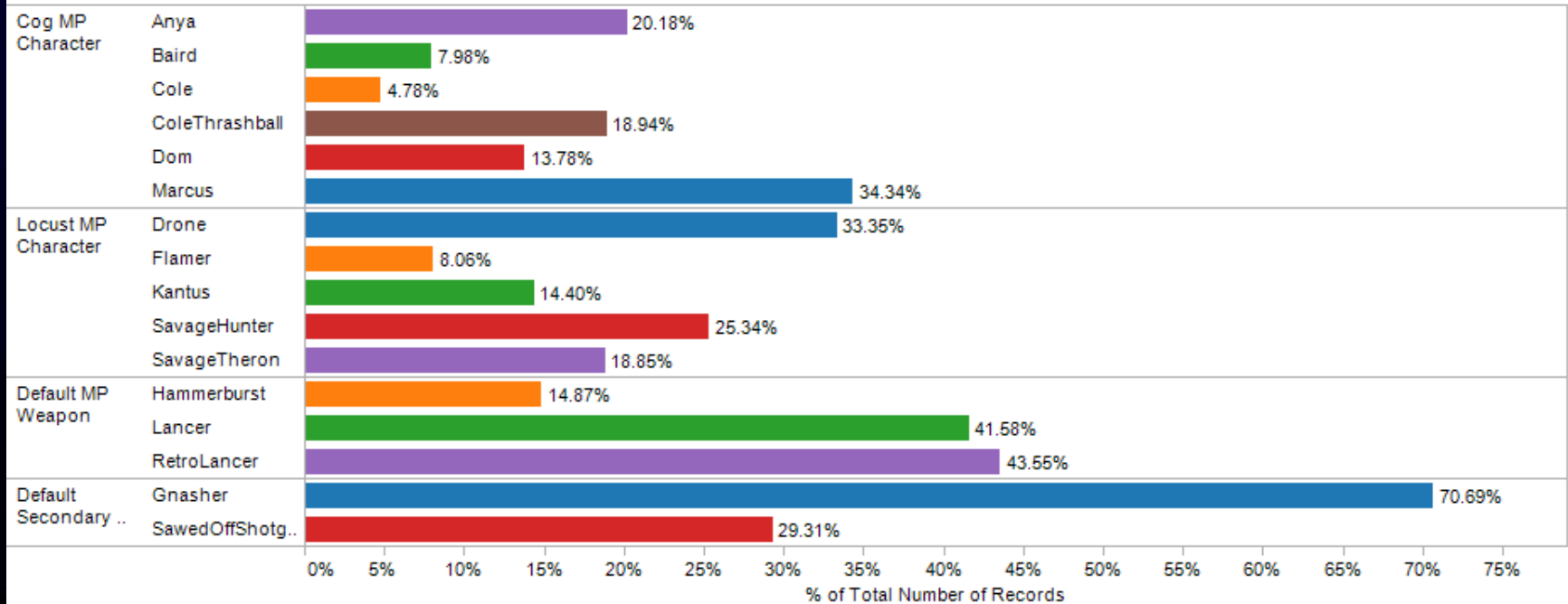About Tableau maps www.tableausoftware.com/mapdata

Total Users

1,289,329

Top Countries by User

Country	Users
UNITED STATES	891,370
UNITED KINGDOM	132,618
MEXICO	54,814
CANADA	52,602
FRANCE	35,885
SPAIN	22,160
AUSTRALIA	14,680
ITALY	14,147
GERMANY	10,190
IRELAND	6,307

Player Data (cont.)

Multi Player Preferences

Category	Item	%
Cog MP Character	Anya	20.18%
	Baird	7.98%
	Cole	4.78%
	ColeThrashball	18.94%
	Dom	13.78%
	Marcus	34.34%
Locust MP Character	Drone	33.35%
	Flamer	8.06%
	Kantus	14.40%
	SavageHunter	25.34%
	SavageTheron	18.85%
Default MP Weapon	Hammerburst	14.87%
	Lancer	41.58%
	RetroLancer	43.55%
Default Secondary ..	Gnasher	70.69%
	SawedOffShotg..	29.31%

% of Total Number of Records

Player Data (cont.)

- NAT (network address table) types
  - Data is different from Gears 2 and final product
  - More Open NATs, fewer Moderate or Strict

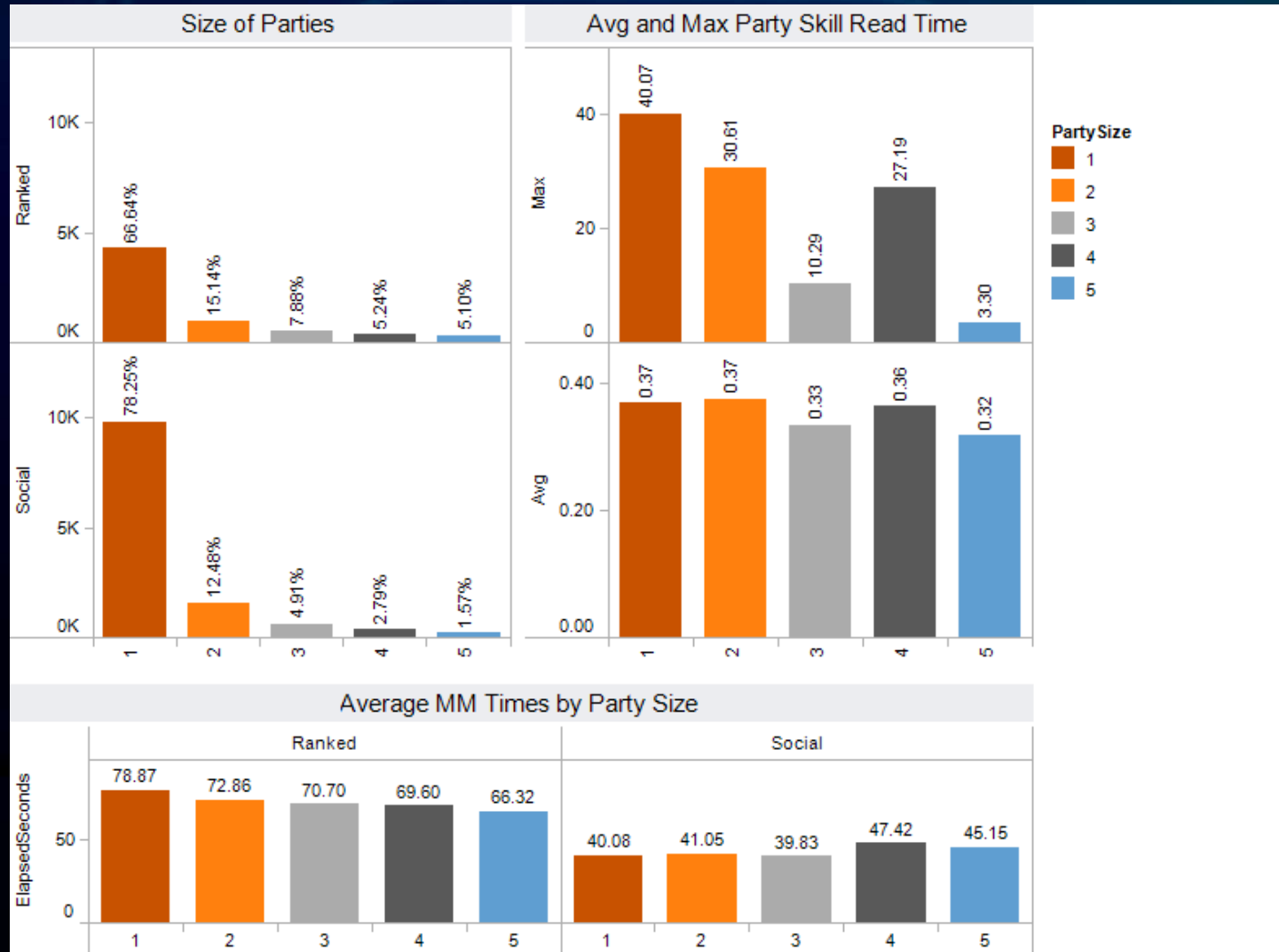|		
Open	████████████████████	79.93%
Moderate	███	16.12%
Strict	█	3.96%
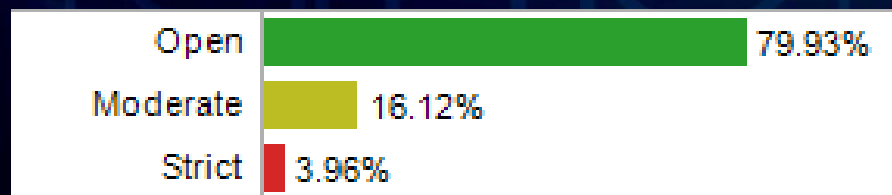
Matchmaking Data

- Search times for each type of search
- Party details
  - Size, unique ids, average skill, average XP level
- Returned host data
  - Ping, match quality, average skill
- Number of searches
- Number of results per search

Matchmaking Data
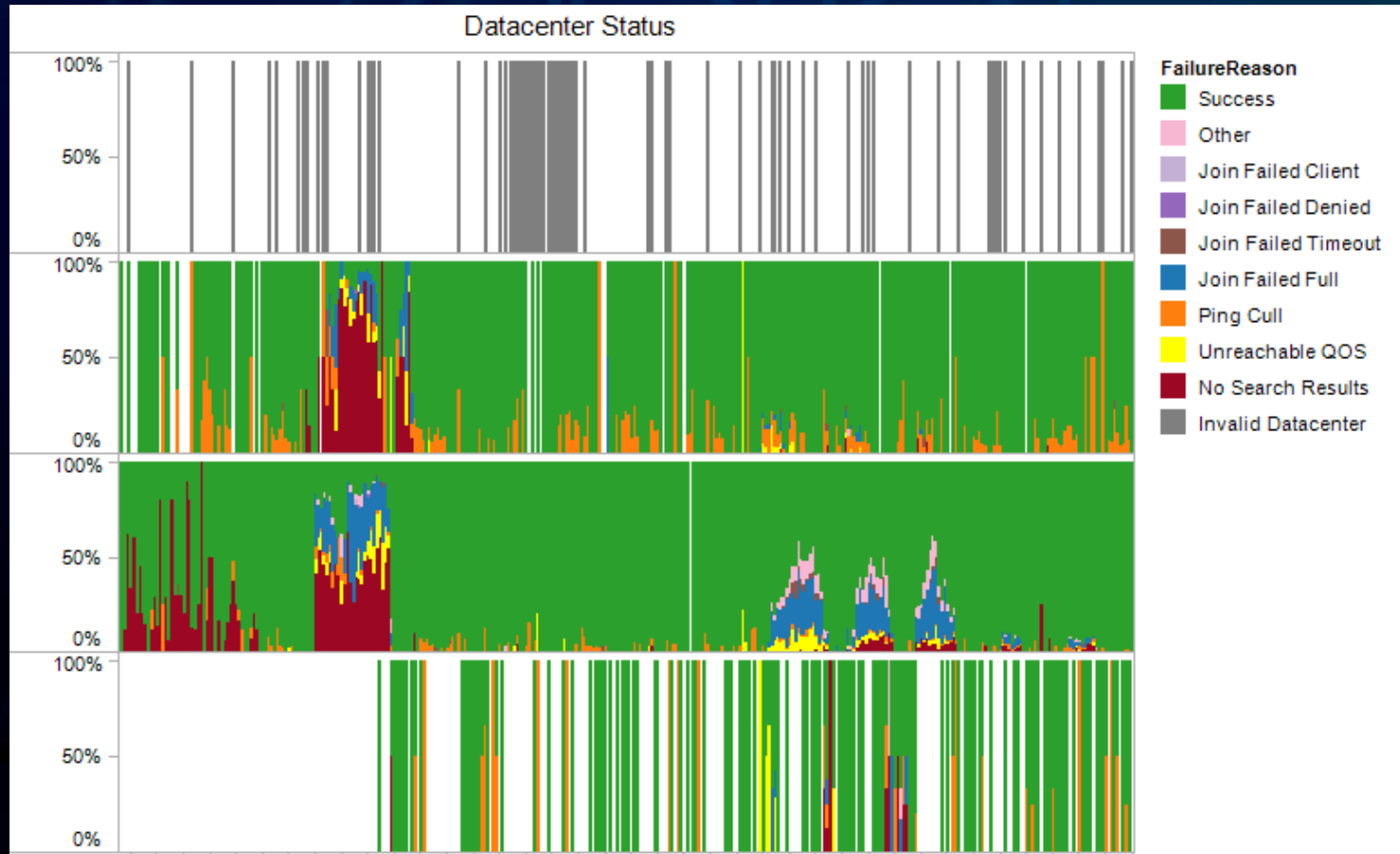
Matchmaking Data (cont.)

Matchmaking Data (cont.)

Gameplay Data

- Map and game mode played
- Weapon data
  - Fired, melee, picked up, dropped
- Player locations recorded for all events
- Each recorded event has a granularity level for throttling via Unreal MCP
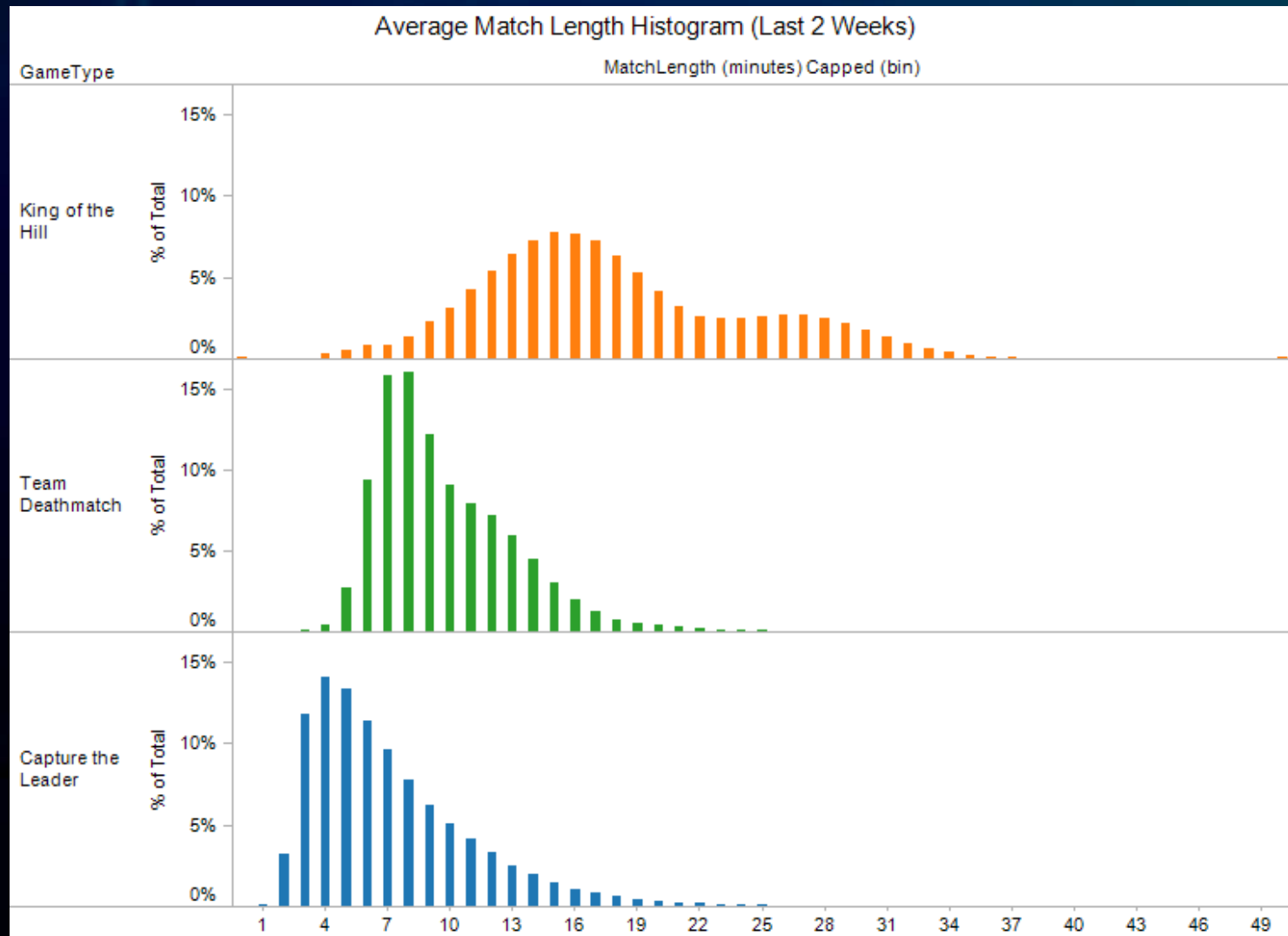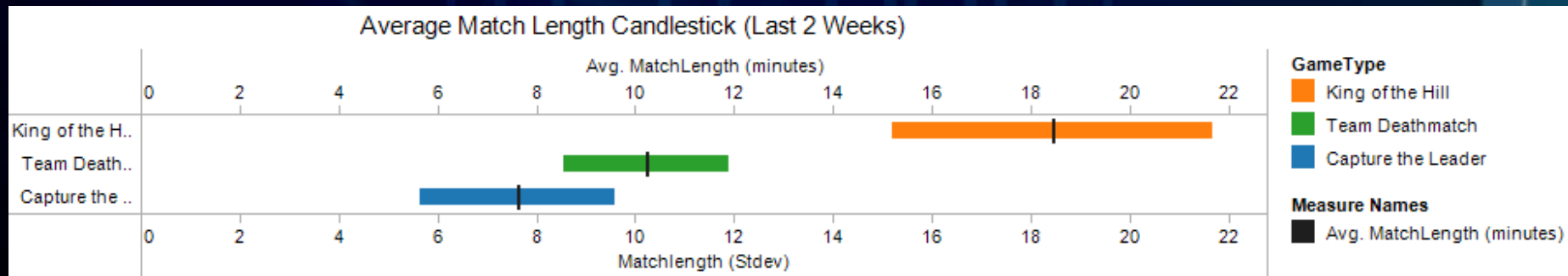
Gameplay Data
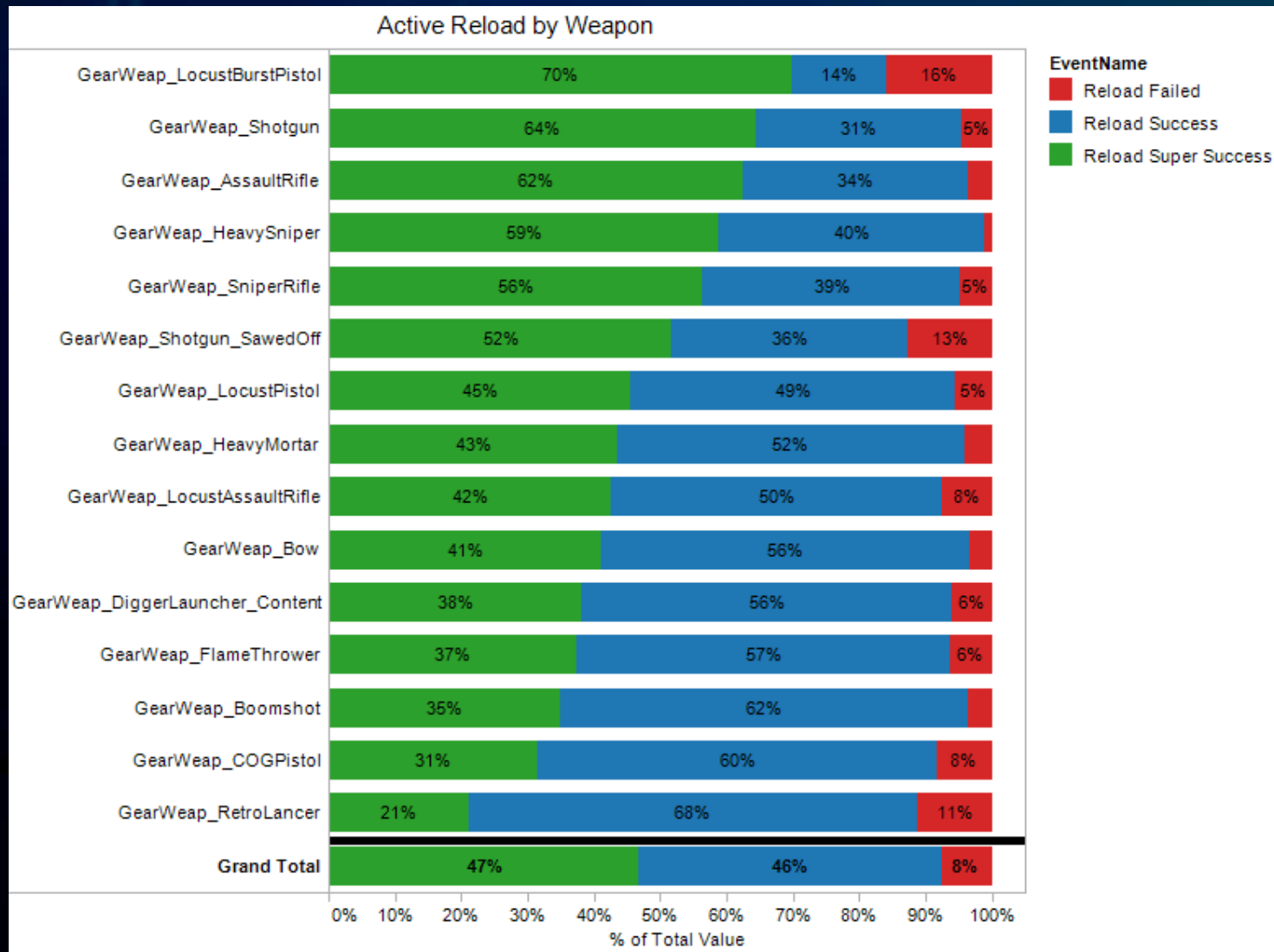
Gameplay Data (cont.)

Gameplay Data (cont.)

Average Match Length Histogram (Last 2 Weeks)
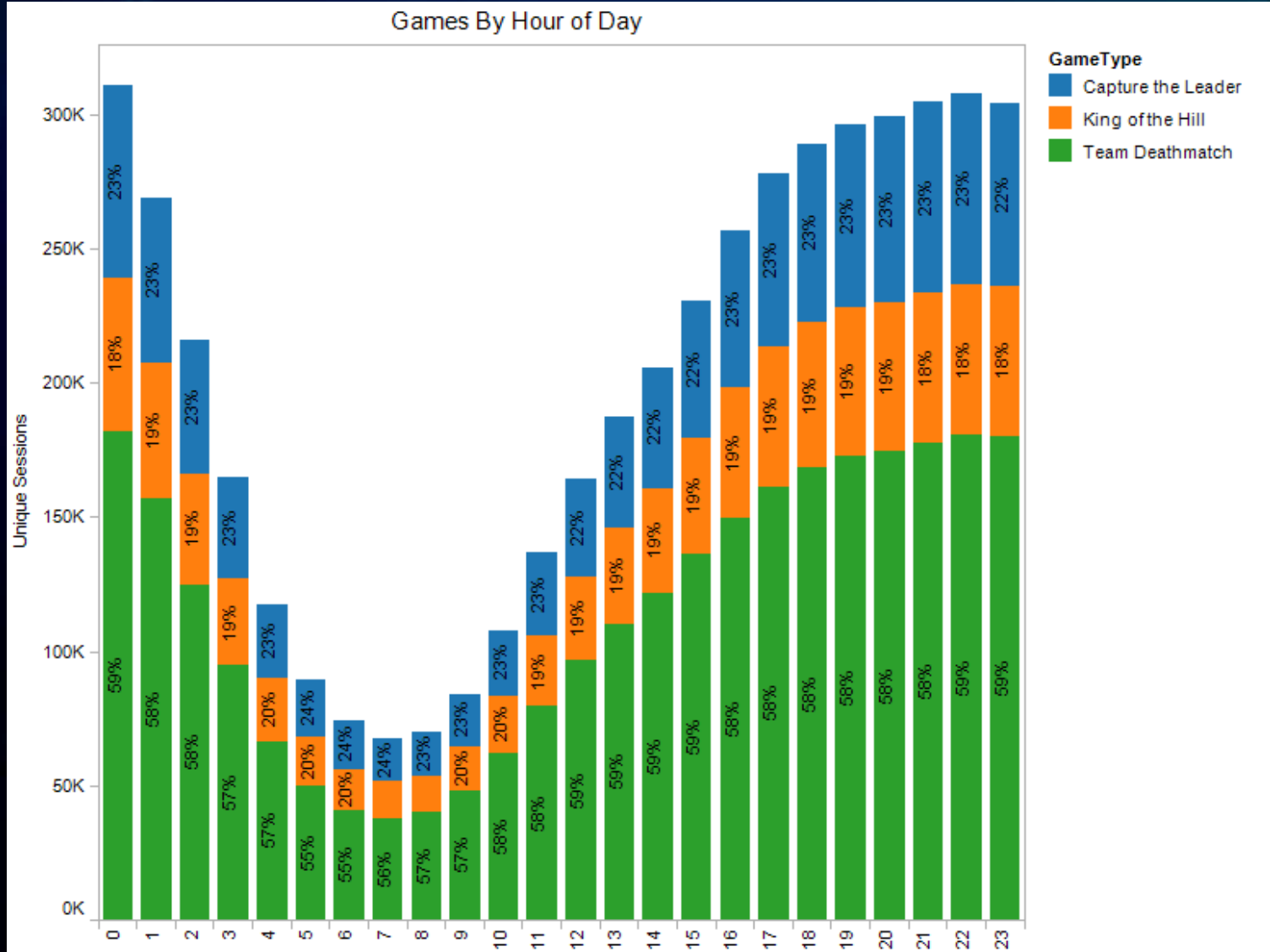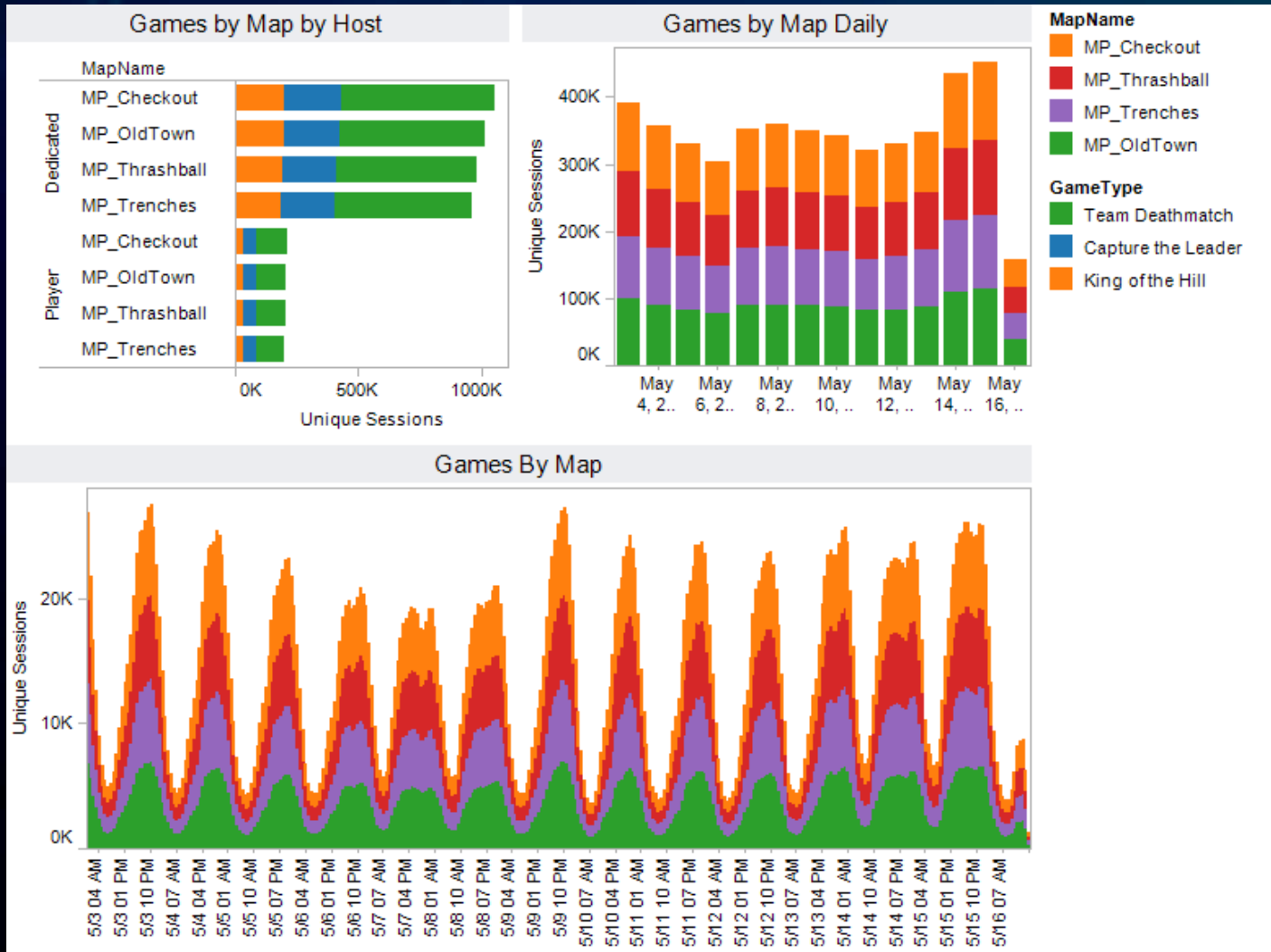
Gameplay Data (cont.)

- Goal was 10 minutes per match
  - Needed to tweak game rules to bring them closer to that target

Average Match Length Candlestick (Last 2 Weeks)

Gameplay Data (cont.)

Active Reload by Weapon
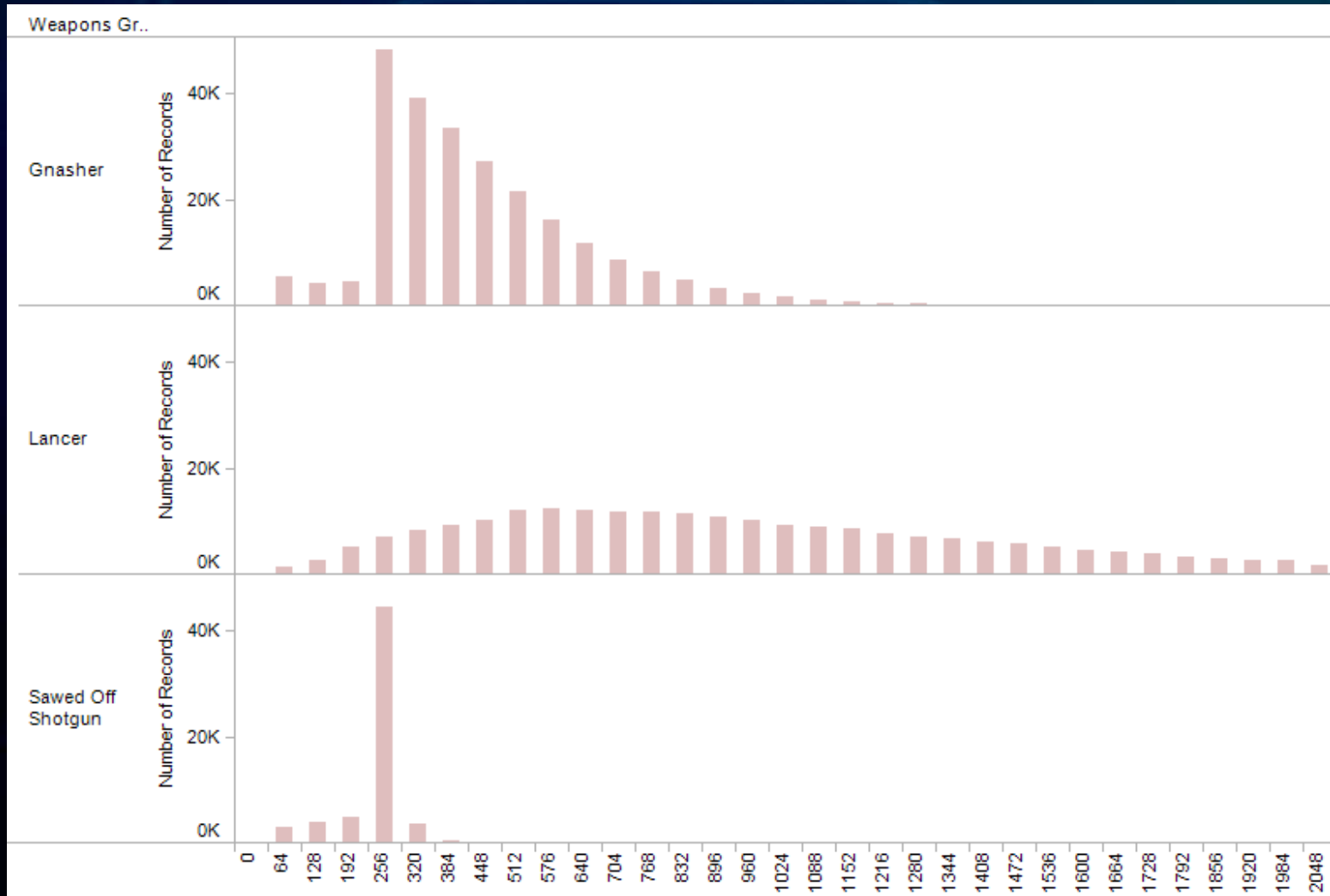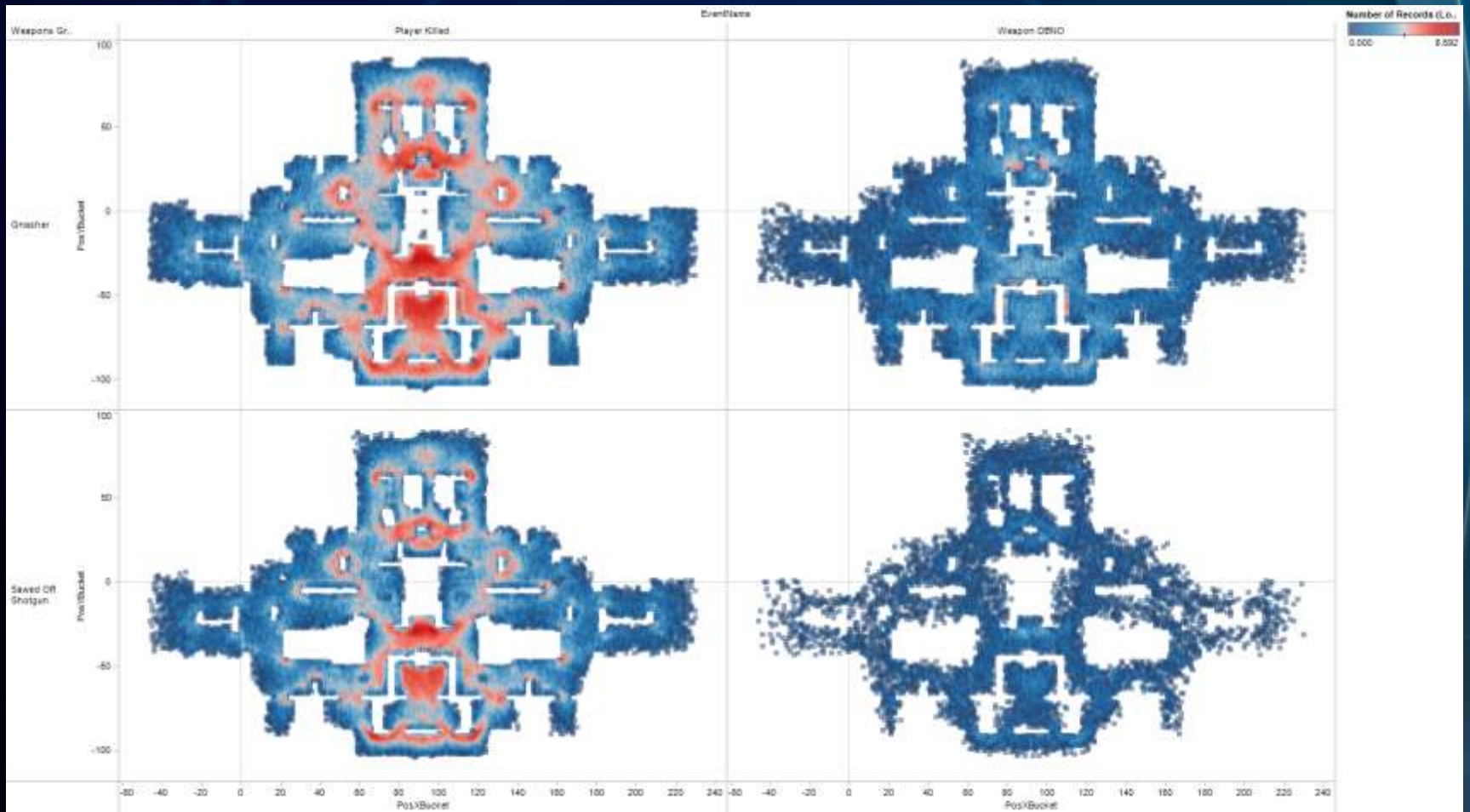
Gameplay Data (cont.)

Gameplay Data (cont.)

Gears Matchmaking

- Gears 3 used the same algorithm as Gears 2
  - Minor changes to accommodate dedicated servers

- Player hosted matches are still required
  - Things can and will go wrong
    - Player's are generally not network engineers
    - Servers can fail and network outages can prevent access to datacenter
    - Some regions have poor infrastructure

Terminology

- Match Quality
  - How likely a match ends in a draw
- QoS – Quality of Service
  - Network quality between client and host
- Best search
  - Exact match based upon party size
- Any search
  - Match based upon space greater than party size
- Empty search
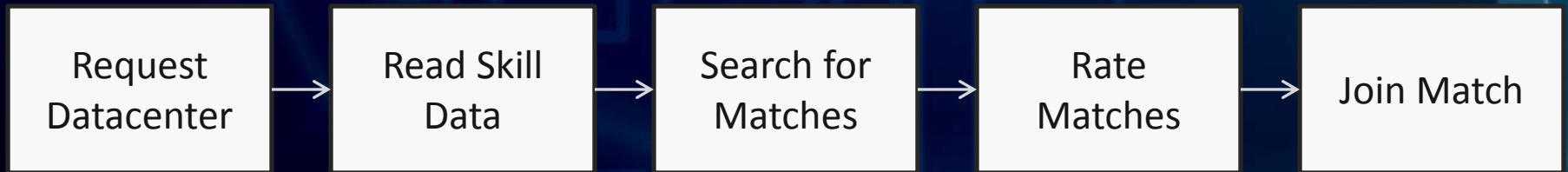 - Find a new server to configure for the desired mode
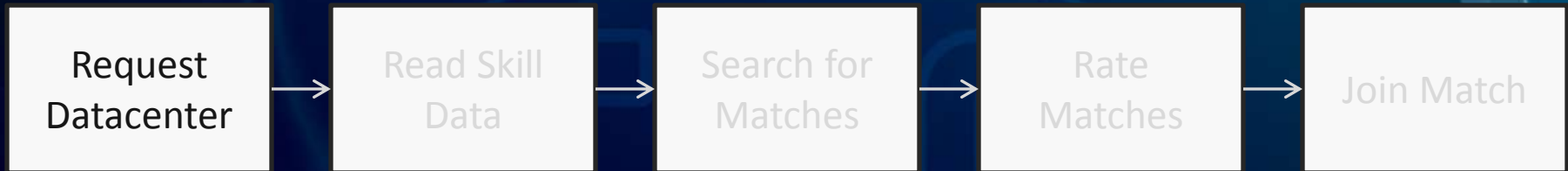
Matchmaking Goals

Matchmaking Time
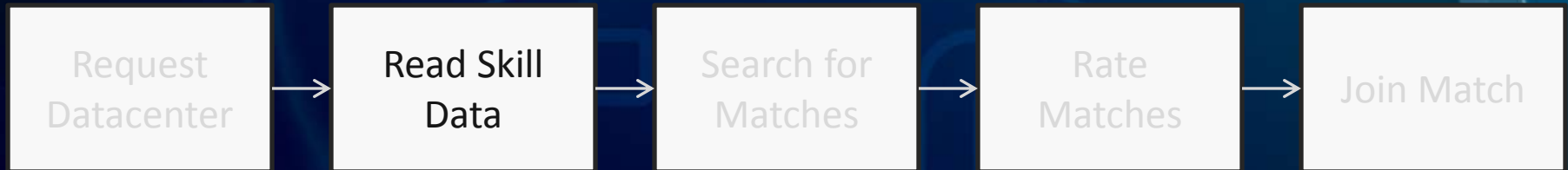
Match Quality

Network Quality

Matchmaking Process

Request Datacenter → Read Skill Data → Search for Matches → Rate Matches → Join Match

Matchmaking Process (cont.)

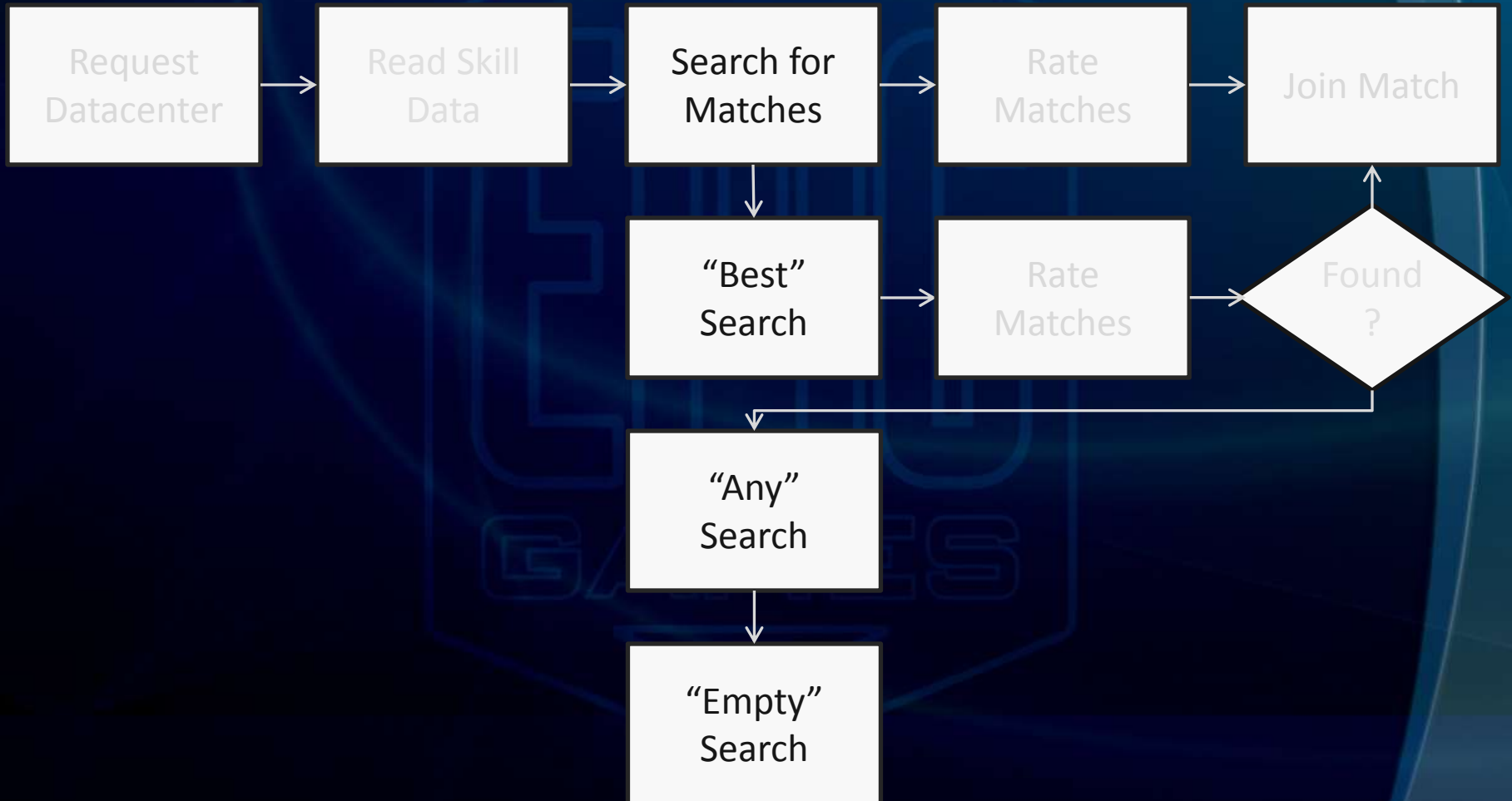| Request Datacenter	→	Read Skill Data	→	Search for Matches	→	Rate Matches	→	Join Match

- Use IP to location to find closest datacenter
  - Unreal MCP returns an ID for this datacenter
- Done once per player signin
- Saved in the player's profile in case the look up fails

Matchmaking Process (cont.)

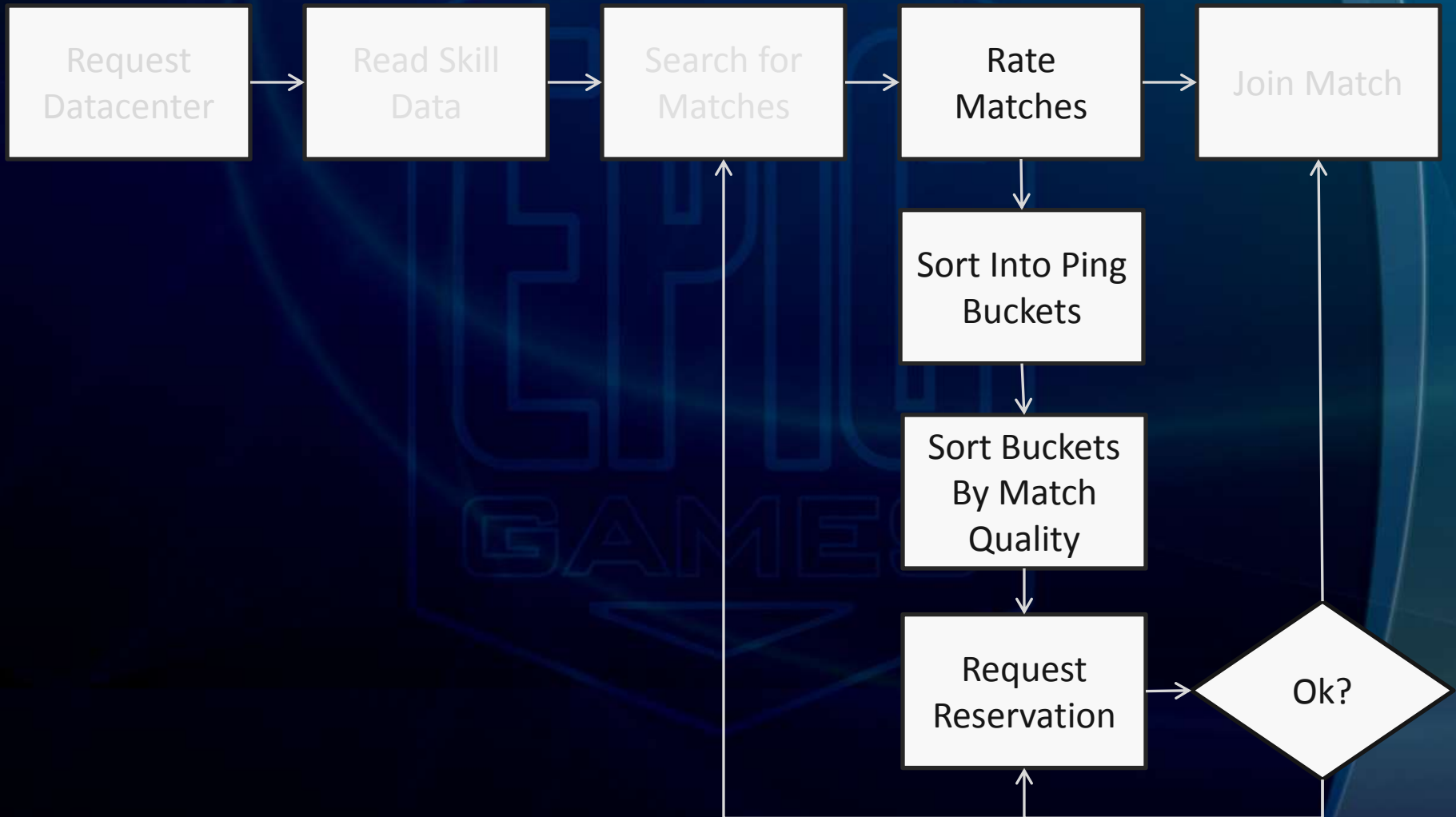| Request Datacenter | → | Read Skill Data | → | Search for Matches | → | Rate Matches | → | Join Match |

- Each playlist has its own skill table
- Done once per search initiation
- Used to provide Xbox Live with a single value to rate advertised sessions against
- Used to calculate match quality

Matchmaking Process (cont.)

Matchmaking Process (cont.)

Request Datacenter → Read Skill Data → Search for Matches → Rate Matches → Join Match

Rate Matches → Sort Into Ping Buckets → Sort Buckets By Match Quality → Request Reservation → Ok?

Beta Launch Plan

- Access to beta was in multiple phases
  - Epic employees, Microsoft employees, friends and family
  - Bulletstorm Epic Edition purchasers
  - Gears of War Pre-orders

- Phased approach allowed us to manage player population growth
  - Thousands, tens of thousands, hundreds of thousands

Beta Launch

- Employees, friends and family
  - Everything is fine

- Bulletstorm players join in
  - Forums and Twitterverse indicate matchmaking times are slow

- War Room
  - We gather to go over the data and drill into what's going on

QoS Data Indicates a Problem

# Too Many Servers Are Unreachable

Leading To Slow Matchmaking

What's Going On?

- Contention on QoS results
  - Dedicated servers were configured like Xbox clients
  - Except all clients were funneled to similar sets of servers: different pattern from player hosted

- The fix?
  - Increase the configurable max number of simultaneous QoS probes
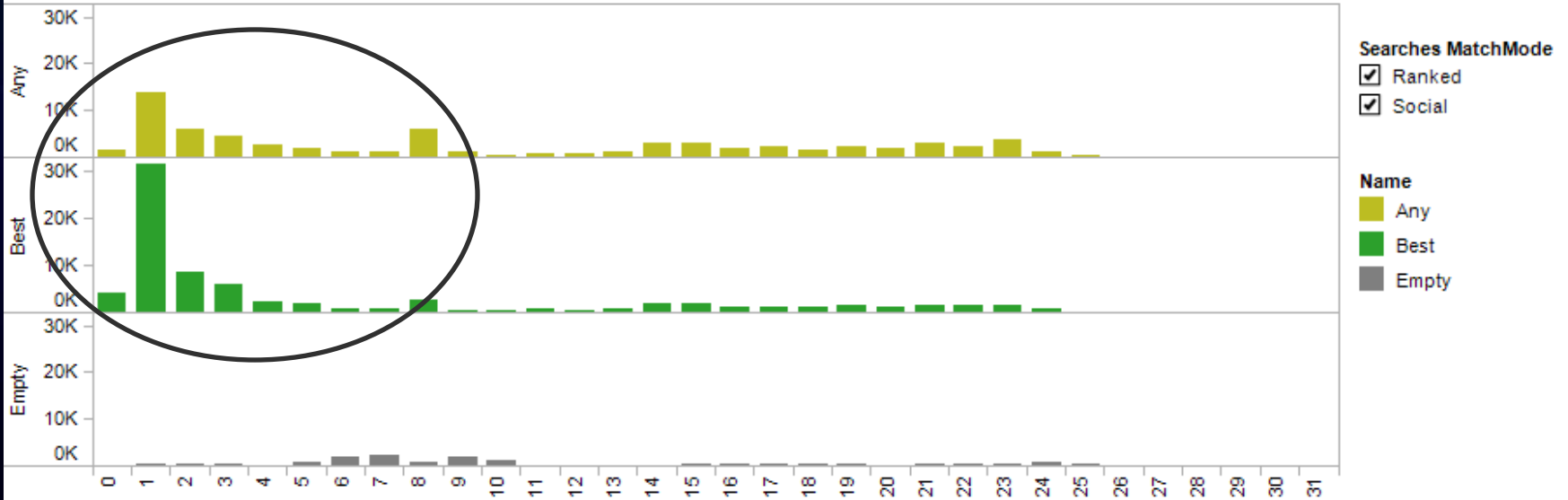
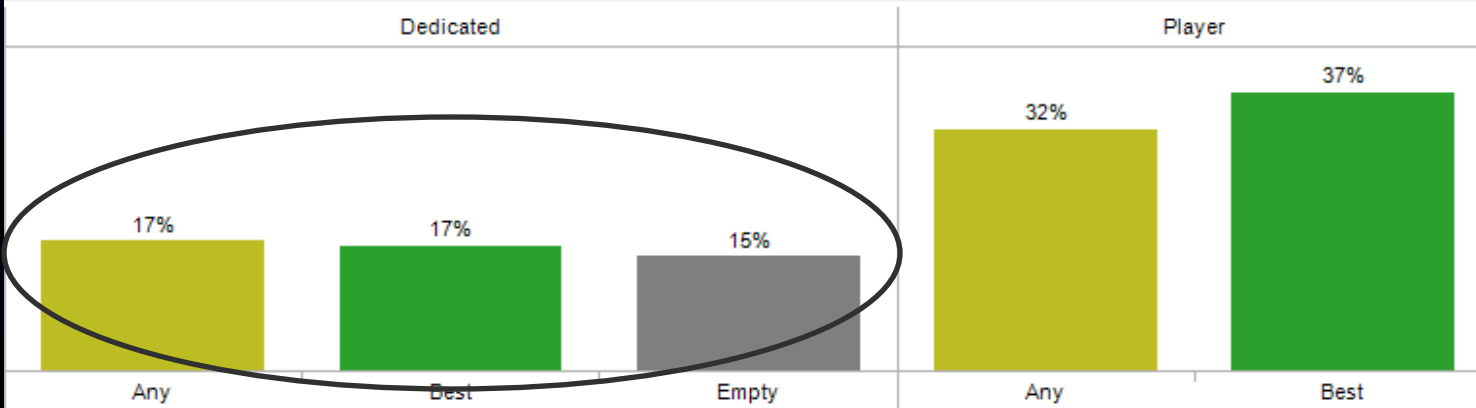# QoS Problem Resolved

QoS Problem Over Time
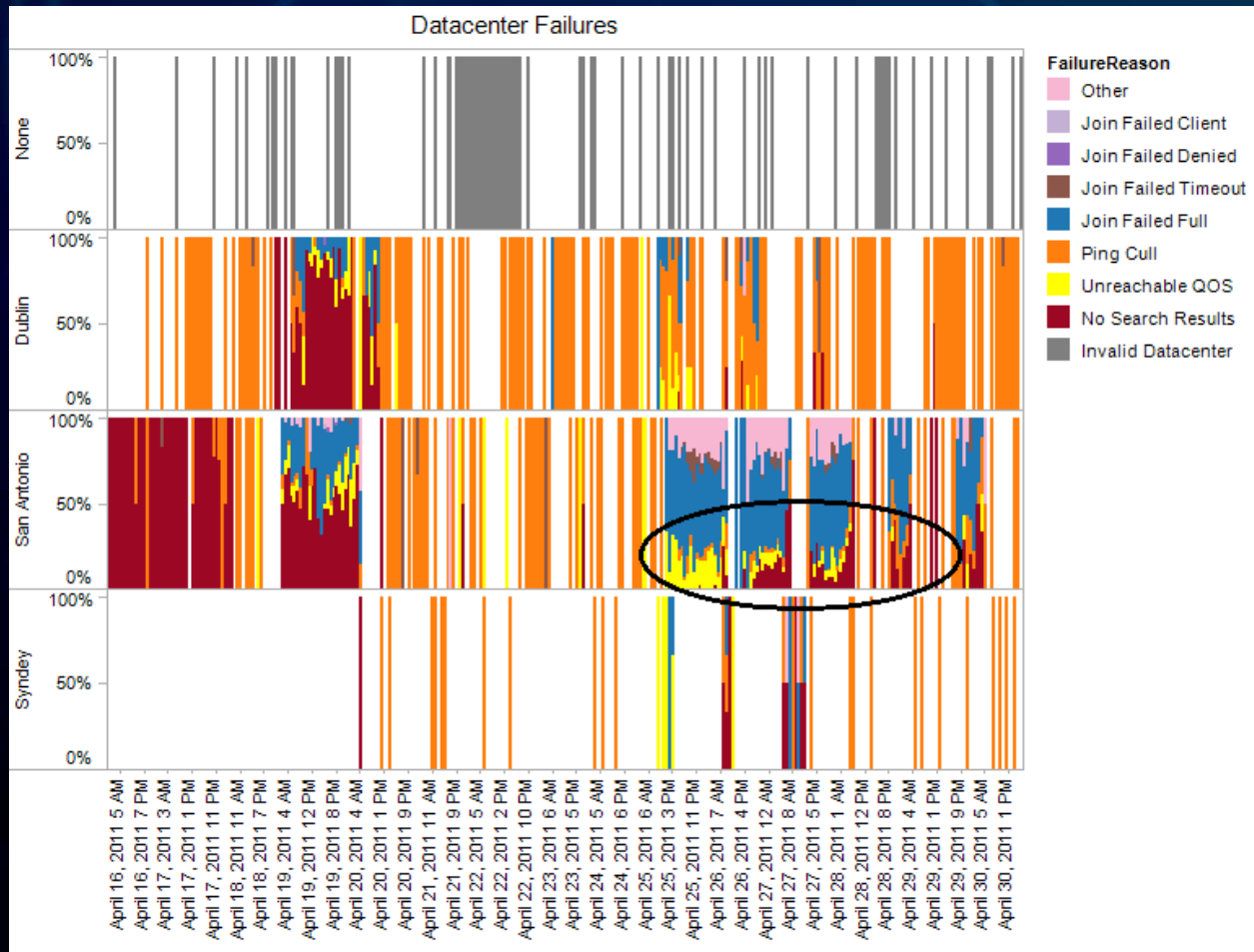
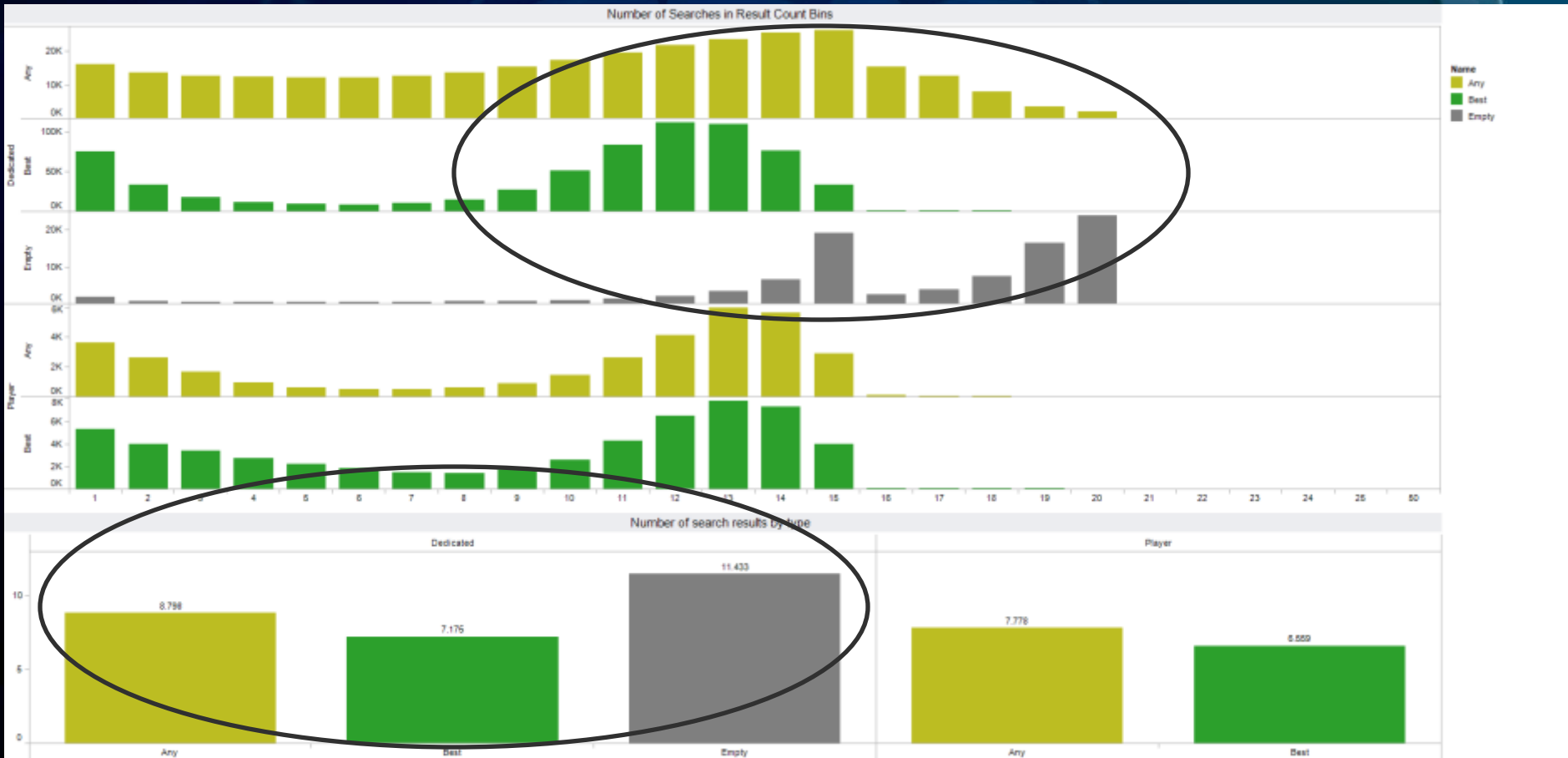Datacenter Failures

Production QoS Data

QoS was improved, but…

- Matchmaking charts show clients are host starved
  - Not enough hosts are being found
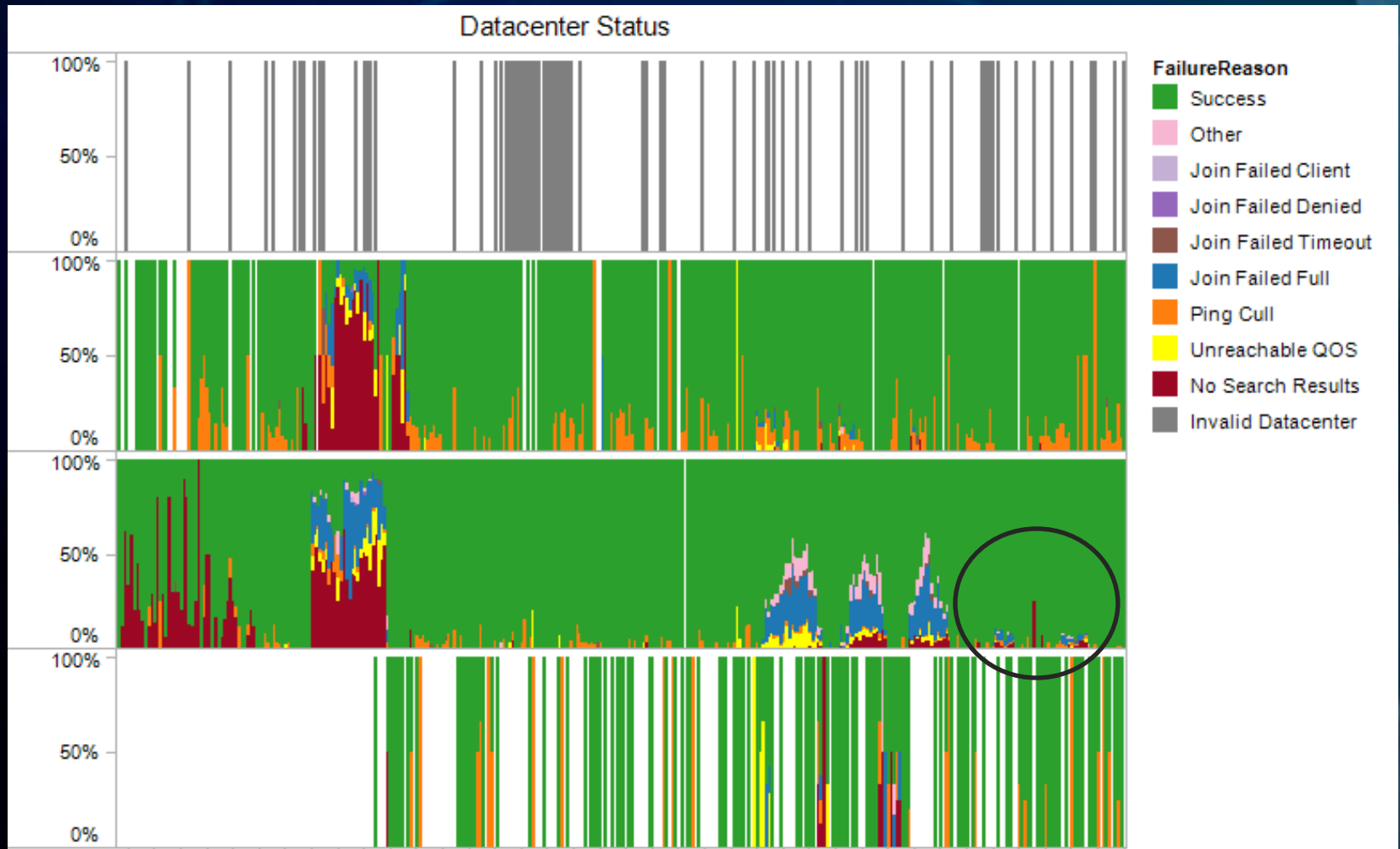 - Resulting in player hosted sessions

Too Few Search Results

Less is More

QoS and Search Results Change

QoS and Search Results Change

Matchmaking Algorithm Experiment

- Prefer matches by distance from party's average XP Level
  - Sorting has to be on the backend where the data is available
  - Risky, because it overrides all standard Xbox Live sorting mechanisms

- Why do this?
  - Player perception is XP Level means skill

Experiment Results

- `<SearchResult ... Ping="50" Quality="0.924" />`
  `<SearchResult ... Ping="100" Quality="0.668" />`
  `<SearchResult ... Ping="100" Quality="0.409" />`

- `<Reservation>`
  `    <Player ... Skill="23" XpLevel="92" />`
  `    <Player ... Skill="28" XpLevel="89" />`
  `    <Player ... Skill="25" XpLevel="86" />`
  `</Reservation>`
  `<Reservation>`
  `    <Player ... Skill="19" XpLevel="90" />`
  `</Reservation>`
  `<Reservation>`
  `    <Player ... Skill="28" XpLevel="87" />`
  `</Reservation>`

Experiment Results

- ```
  <SearchResult … Ping="50" Quality="0.924" />
  <SearchResult … Ping="100" Quality="0.668" />
  <SearchResult … Ping="100" Quality="0.409" />
  ```

- ```
  <Reservation>
       <Player … Skill="23" XpLevel="92" />
       <Player … Skill="28" XpLevel="89" />
       <Player … Skill="25" XpLevel="86" />
  </Reservation>
  <Reservation>
       <Player … Skill="19" XpLevel="90" />
  </Reservation>
  <Reservation>
       <Player … Skill="28" XpLevel="87" />
  </Reservation>
  ```

Experiment Results

- `<SearchResult ... Ping="50" Quality="0.924" />`
  `<SearchResult ... Ping="100" Quality="0.668" />`
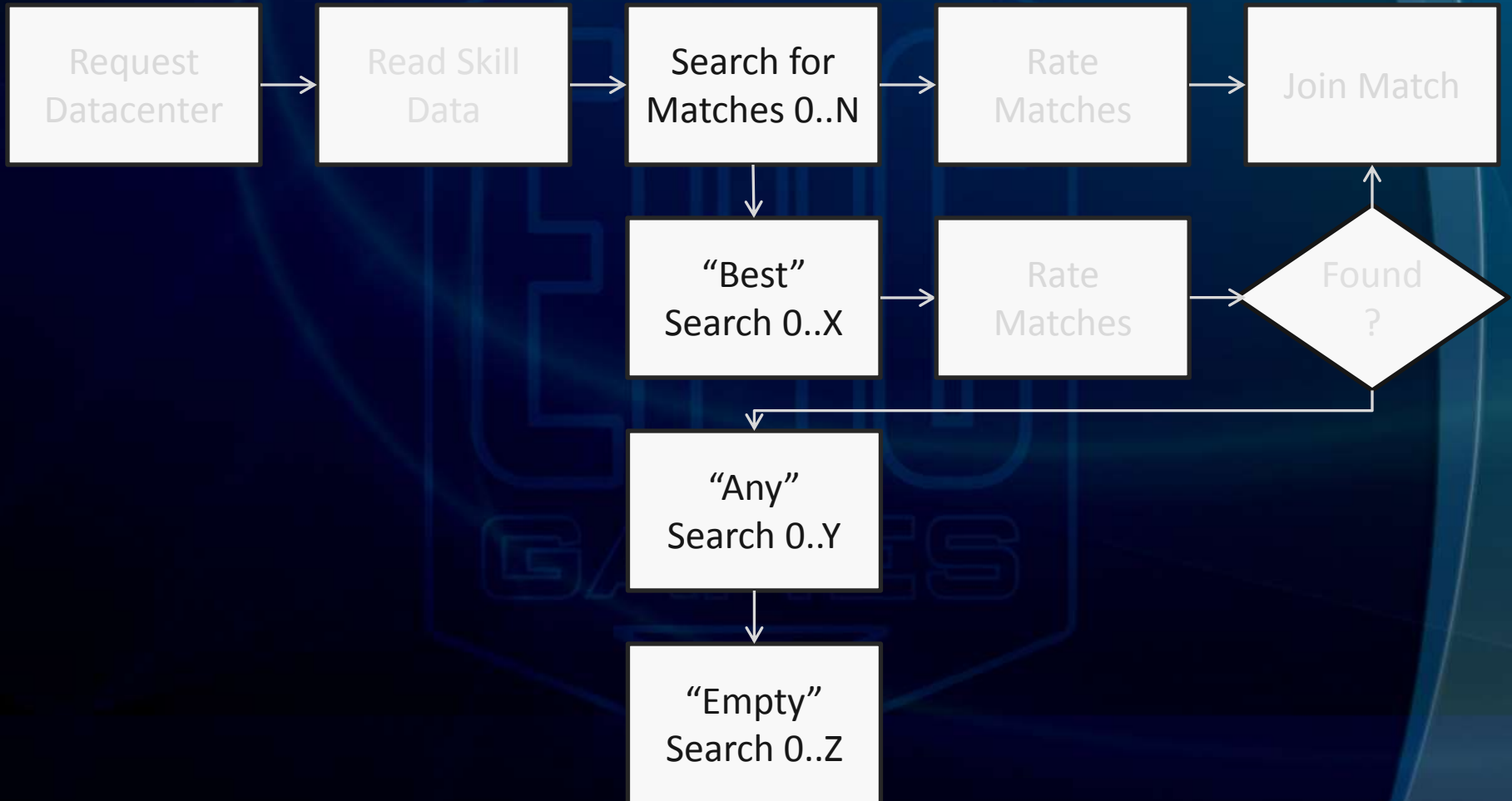 `<SearchResult ... Ping="100" Quality="0.409" />`

- `<Reservation>`
  `    <Player ... Skill="23" XpLevel="92" />`
  `    <Player ... Skill="28" XpLevel="89" />`
  `    <Player ... Skill="25" XpLevel="86" />`
  `</Reservation>`
  `<Reservation>`
  `    <Player ... Skill="19" XpLevel="90" />`
  `</Reservation>`
  `<Reservation>`
  `    <Player ... Skill="28" XpLevel="87" />`
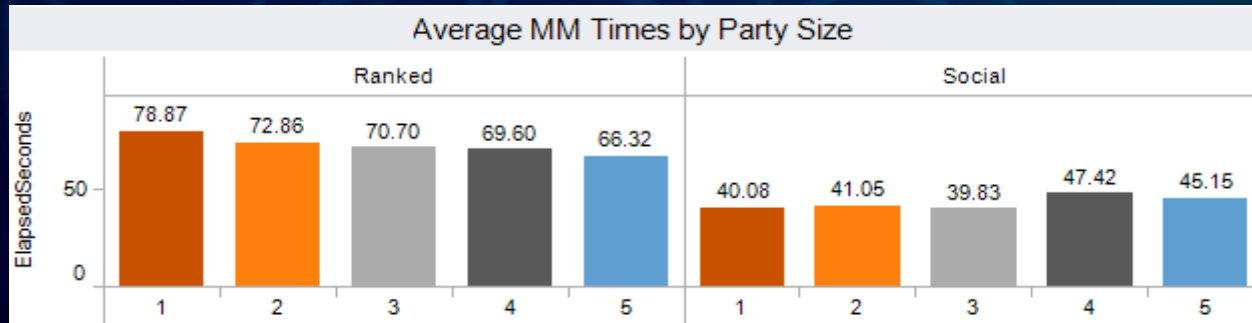 `</Reservation>`

Beta Results

Production Changes

- Added the ability to place servers in buckets

- Added looping in addition to bucketing

- Added new QoS configuration options

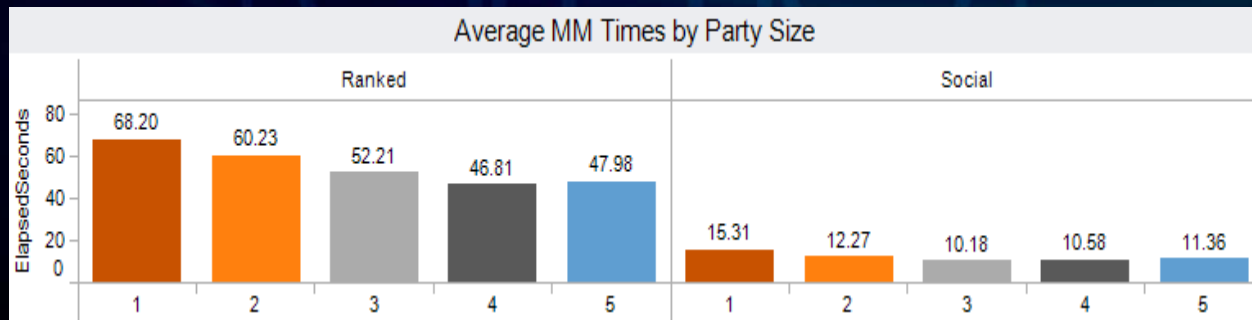- Changed Quick Match searching to be XP Level based

Production Matchmaking Process

Matchmaking Times

Beta

Production

Takeaways

- Record all input matchmaking variables

- Record all output results
  - Don't assume a detail isn't important

- You can never have too many configurable knobs to turn
  - We couldn't change some values in the beta

Why I learned to love the beta

- The extra effort up front was offset by removing the need for emergency patches

- Instead of going from 0 to 1 million, we could learn while managing the growth

- We discovered and addressed bottlenecks in every phase of our data collection pipeline
  - The data doesn't do you any good if you can't look at it

- Gears of War 3 launch was our best ever

Gears of War 3 Analytics:

Optimizing the Online Experience, or How I Learned to Stop Worrying and Love the Beta

Special thanks to:

Wes Hunt

Josh Markiewicz

Justin Rowland

Ian Thomas

Sam Zamani

Joe Graf

Epic Games, Inc.

Twitter: @EpicCog