
UBM
TechWeb
MMO 101:
Building Disney's Server Systems
GDC Online
Game Developers Conference® Online
October 5-8, 2010 | Austin, TX
Visit www.GDCOnline.com for more information

MMO 101:

Approaches that have been taken at Disney Online Studios in the development of our MMO environments.

Roger H. Hughston

- **Almost 25 years as a computer professional.**
- **Over 20 years with Disney.**
  - **Information Technology / Imagineering / VR studio / Disney Online.**
- **Disney Online Studios - Director of Architecture, Research & Development.**
- **A bunch of MMO and MMO type projects.**
- **Love to play games and love to build them.**

OK, what is a Disney MMO?

- A computer experience in which a large number of children and their families can simultaneously interact in a persistent world.

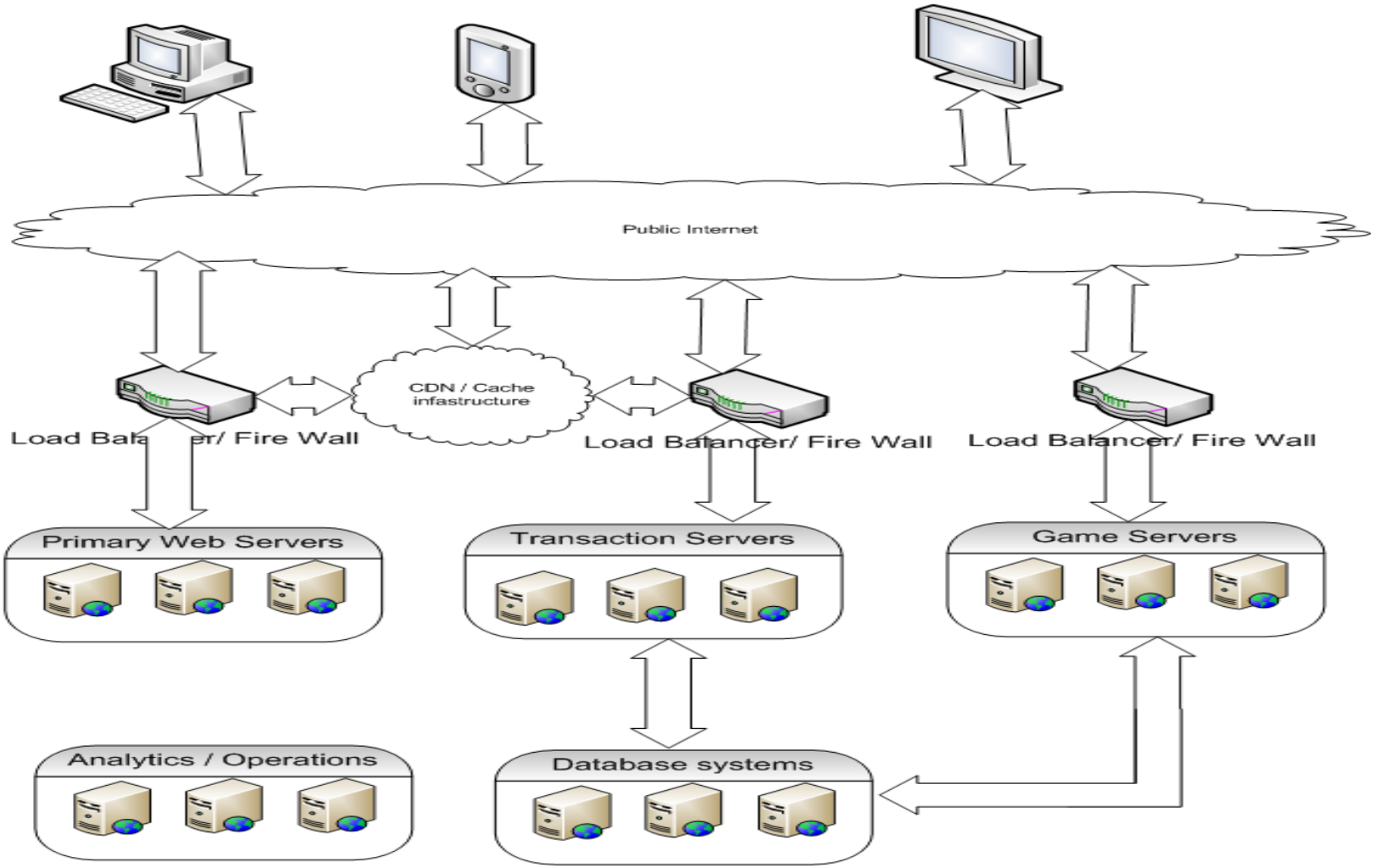
- Children and their families
- Large number of
- Simultaneous
- Interactive
- Persistent

I'm a geek and this a Technical Talk!

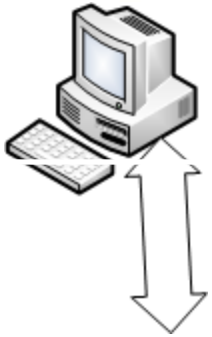
- Talk about the technologies used.
- Talk about the hard problems
- Talk about a game server topology.

- Technical - Approaches that have been taken at Disney Online Studios in the development of our MMO environments.

A typical Disney environment



Game Clients



Disney CLUB PENGUIN™ Flash AS2–Browser–Web Assets

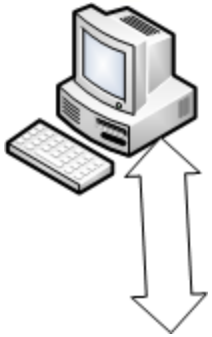
Disney's TOONTOWN ONLINE℠ Panda 3D–Browser/Fat Client–Web/Local Assets

Disney fairies Pixie Hollow Flash AS3–Browser–Web Assets

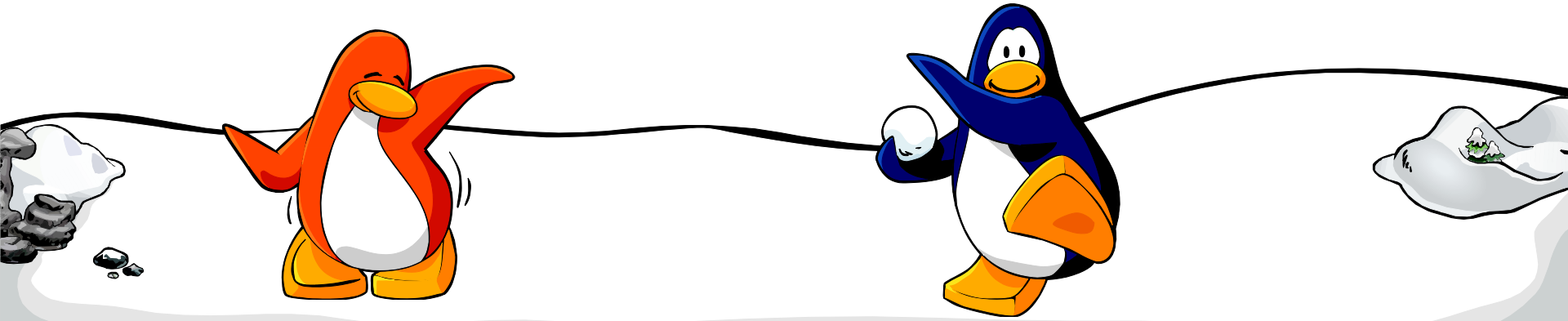
Disney PIRATES of the CARIBBEAN ONLINE Panda3D-Browser/Fat Client–Web/Local Assets

THE WORLD OF Cars ONLINE Flash AS3–Browser–Web Assets

Client Communications

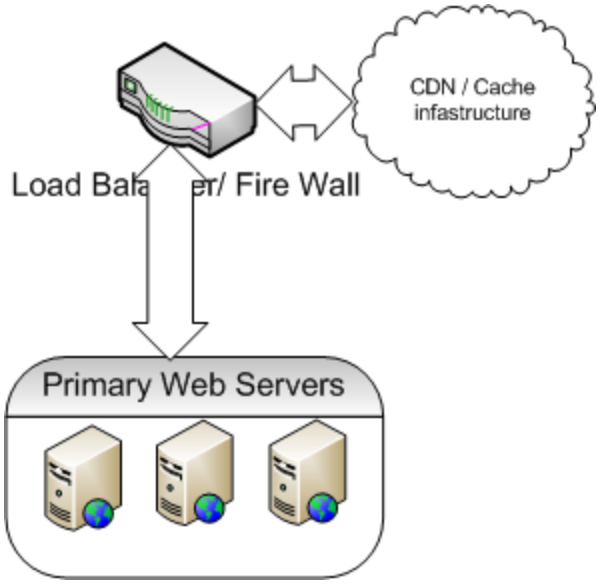


- HTTP(s)
  - Preferred transport. Decoupled and stateless.
- None HTTP = TCP.
  - We like the guaranties we get with TCP.
  - Accessibility is higher priority than latency.



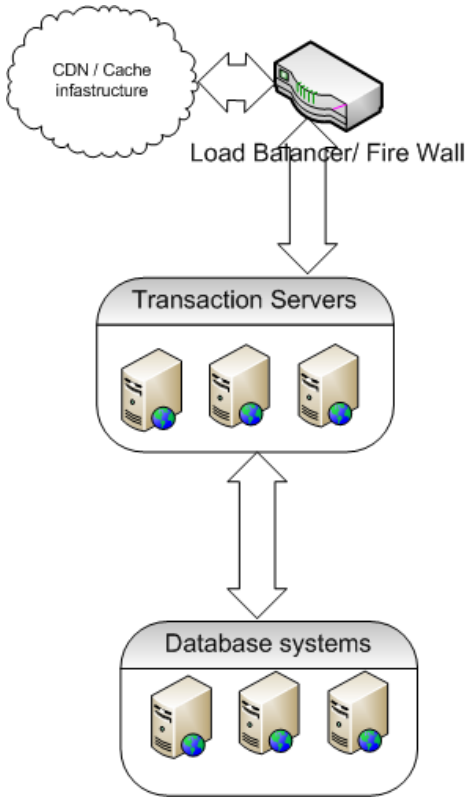
Primary HTTP Server Farms

- HTTP is the first choice for infrastructure!
  - Load balance for scaling and resiliency
  - Cache are your friends.
  - HTTP is a commodity.
  - The world is optimizing.
    - PNG, SWFT, HTML, Patching files…
    - Streaming content and code…
    - 0 to N of these farms…
    - Linux, Apache, Squid, CDN's



Transaction Server Farms

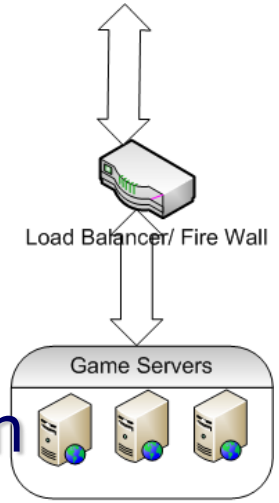
- HTTP is the choice for low fidelity transactions.
  - HTTP is a commodity.

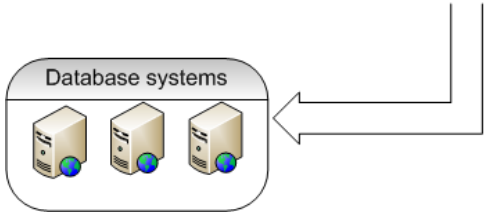


- Login, purchases, profiles, blogs…
- Leader boards, achievements…
  - 0 to N of these farms…
  - Linux, Apache, Tomcat, PHP, Java

Game Server Clusters

- A series of machines and data storage used to manage the game logic.
  - Custom servers and server clusters.
  - Smart FOX servers – Heavy use of custom extensions.



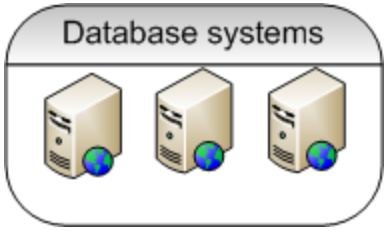


Linux, Java, c++, Python

Database Systems

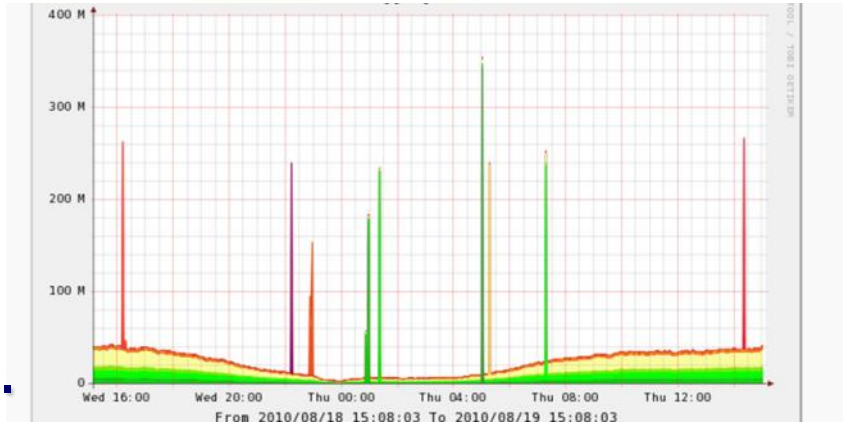
- Persistent storage.
- Low bandwidth shared data.
- Hot Redundancy.

- 1 to N per environment.



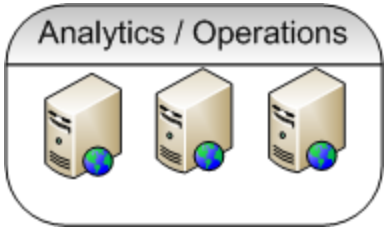
- MySQL, Oracle, Custom…

Instrument servers



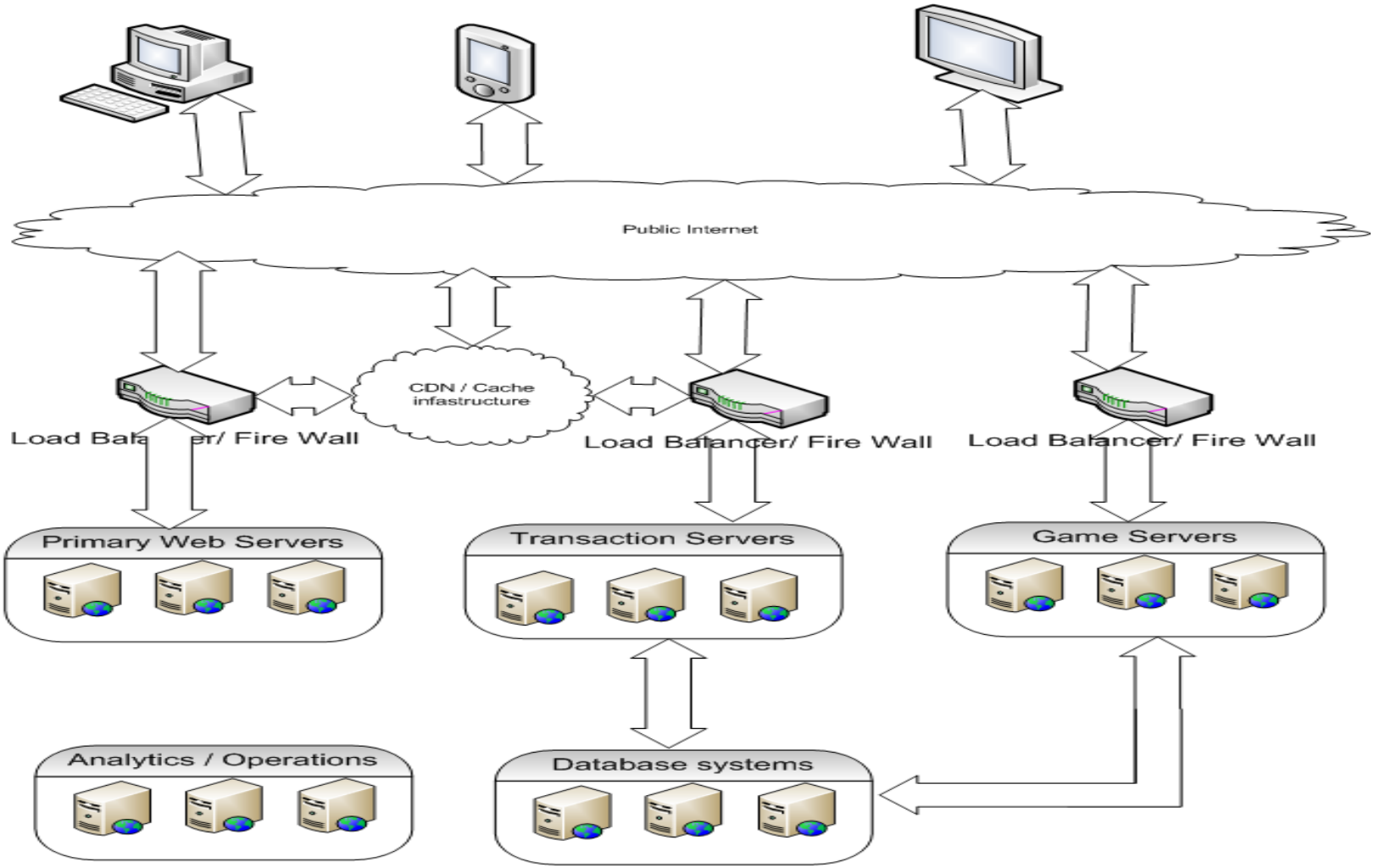
- Record everything.
  - Game events and Accumulators.
  - OS/ Network/ disk... events and Accumulators.
  - Business event and accumulators.

  - When you think you have too much - add more.



  - Real-time access simplifies problem resolution.

A typical Disney environment



Let's talk about the hard problems.

- Data - Rate of change
- Data - Latency
- Data - Size
- Scaling
- Security
- Development
- Resiliency

Data - Rate of Change

- The Data changes at a high rate.
  - Up to ~100ms.
    - Clients want this data fast.

Statefull solved with HTTP solutions.

Stateless solved with Push, not a PULL (HTTP).

- Late entry
- Early exit
- Delta Notification

- Adopt the concept of a budget.
  - "Rate of Change is a budget problem."

Data - Size

- You have to move this over the internet.
- You have to move this in memory(s).
- You have to move this to and from disk.

- Size really does matter.

  - "Data Size is a budget problem."

Data - Latency

- Get over it!
  - The internet is distance and distance is latency.
  - Adopt the concept of a Limit.
    - ~100 – 200 ms.
    - This will help other parts of the system and force all engineers to help address its existence.
  - "Data Latency is not a budget problem."

Scaling

- Make it Faster-Optimizing — Vertical
  - Improved hardware and software
- Replicate-Cache — Horizontal
  - Wider hardware and software = Farms.
  - Wider distribution = Cache is our friend.
  - Wider commercial distribution = CDNs.
- Isolation — Depth
  - New disjoined Hardware - software
  - Just use a different……

Not just expansion, contraction is important also!

Security

- Do not trust the…
  - Client
  - Hosting software
  - Environment
  - …
- Ouch. I must trust them ?
  - Trust where you must
  - Verify
  - Audit

Development = Big $$$ and Time

- MMO's are constantly changing
  - Build infrastructure not direct solutions
  - Build protocols not stand alone solutions
  - Prefer lose integration over tight coupling
- Tools Tools Tools
  - Enable the right talent to do their jobs
  - Get other disciplines out of their way
  - Optimize the ownership pipeline
- Allow for Federation!
  - Simple ability to have a lot of environments active at once
  - Only way to reasonably support disjoint development

Resiliency.

Expect things to fail – Plan for it

Build it in.

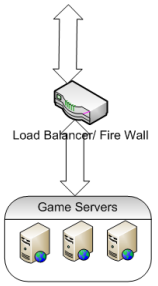
- Prefer Multi Master Model
- Else Replication
- Else Hot Spare
- Else ??

- Fail Soft not Hard!
- Ask early and often – "What happens when this fails?"

The hard problems.

- Have a budget (size, speed)
- Know the Limits (latency, …..)
- Scaling-Horizontal first.
- Security–Trust and verify were you must.
- Resiliency–scale from micro to monster size.
- Dev … d $$
  - Th…

A Game-server Topology



- Distributed Class systems
- Messaging Bus
- Network Culling
  - Rooms/Zones
  - Interest

  - Publish / Subscribe / Messaging

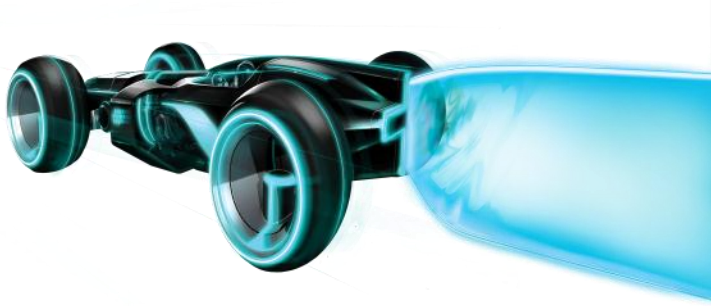
Game-servers and the Clients

- All Worlds all seamlessly accessible.
- Change fidelity is high.
  - Let’s give the client what he needs when he needs it.
    - It’s all about pushing data to a client.
- The data size is very large.
  - Culling and statefull updates are required.

Distributed Class System.

- Most of our languages are class based, Fits very nicely.
- Language agnostic definitions and contracts.
- Contract for serializing, interest reflection, and high level security.

  - Atomic and molecular data types.
  - Data and Function signature and dispatching.
  - Bindings for C++, Java, Python, Action Script, C#...
  - Support for Class instance life cycle.
  - Class asynchronous up calls.
  - High level security and interest routing.



  - Atomic generation, deletion, function dispatching, and document type messaging.

Distributed Class System

```
struct BarrierData {
  uint16 context;
  string name;
  uint32 avIds[];
};

dclass DistributedObject {
  setBarrierData(BarrierData data[]) broadcast ram;
  setBarrierReady(uint16 context[2])  airecv  clsend;
  execCommand(string, uint32 mwMgrId, uint32 avId, uint32 zoneId);
  broadcastMessage() broadcast;
};

dclass DistributedTestObject : DistributedObject
{
  setRequiredField(uint32 r = 78) required broadcast ram;
  setB(uint32 B) broadcast;
  setBA(uint32 BA) broadcast airecv;
  setBO(uint32 BO) broadcast ownsend db;
  setBR(uint32 BR) broadcast ram;
  setBRA(uint32 BRA) broadcast ram airecv;
};
```

Room/Zone Base Divisions

- Container used for grouping.
- Entities/Instances exist in zones.
- Pertinent game data is reflected to the zone observers.
- We map important groupings to zones.
  - Locations in the world.
  - Groups of entities.
  - Simple broadcast multiplexers.
  - …

Simple and Complex Interest

- Allow multiple interests to be active at once.
  - Location of the avatar is in.
  - Location the guild member are listening to.
  - Location all Tinker Fairies listen to.
  - Location of world population records.
- Cut scene transitions.
  - Interest follows the avatar.
  - EOF indicator.
- Smooth zoning transitions.
  - Multiple interest foreshadows the avatar.

Zone + Interest = Network Culling!

- Discovery, Filtering, Targeted groups…
- Implemented with a messaging channel pattern.

- Command message pattern.
- Event messaging patterns.
- Request-reply patterns.

Channels=Phone System in the Cluster

- 64bit .. Usually described as 2x32 bit values.

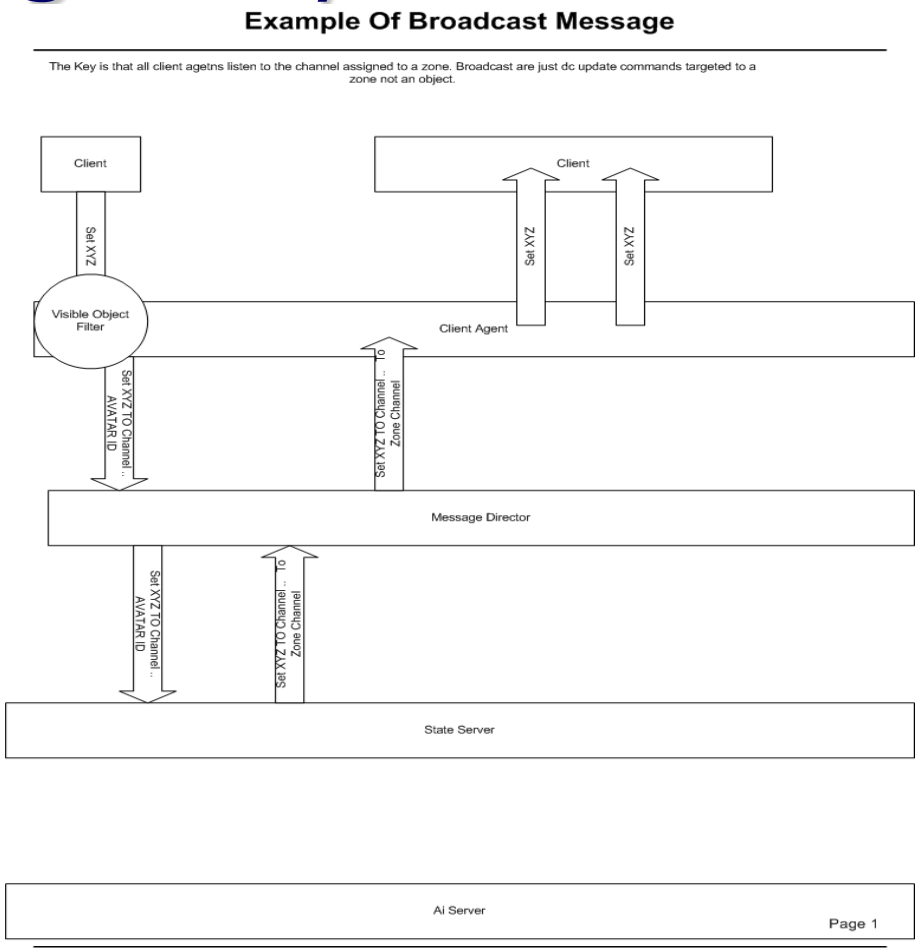
- Point-to-point channels.
- Publish-subscribe channels.
- Data type Channels.
- Interest type channels.

- Game Clients have no access to channels.
- Game Clients are function and DC driven only. (security)

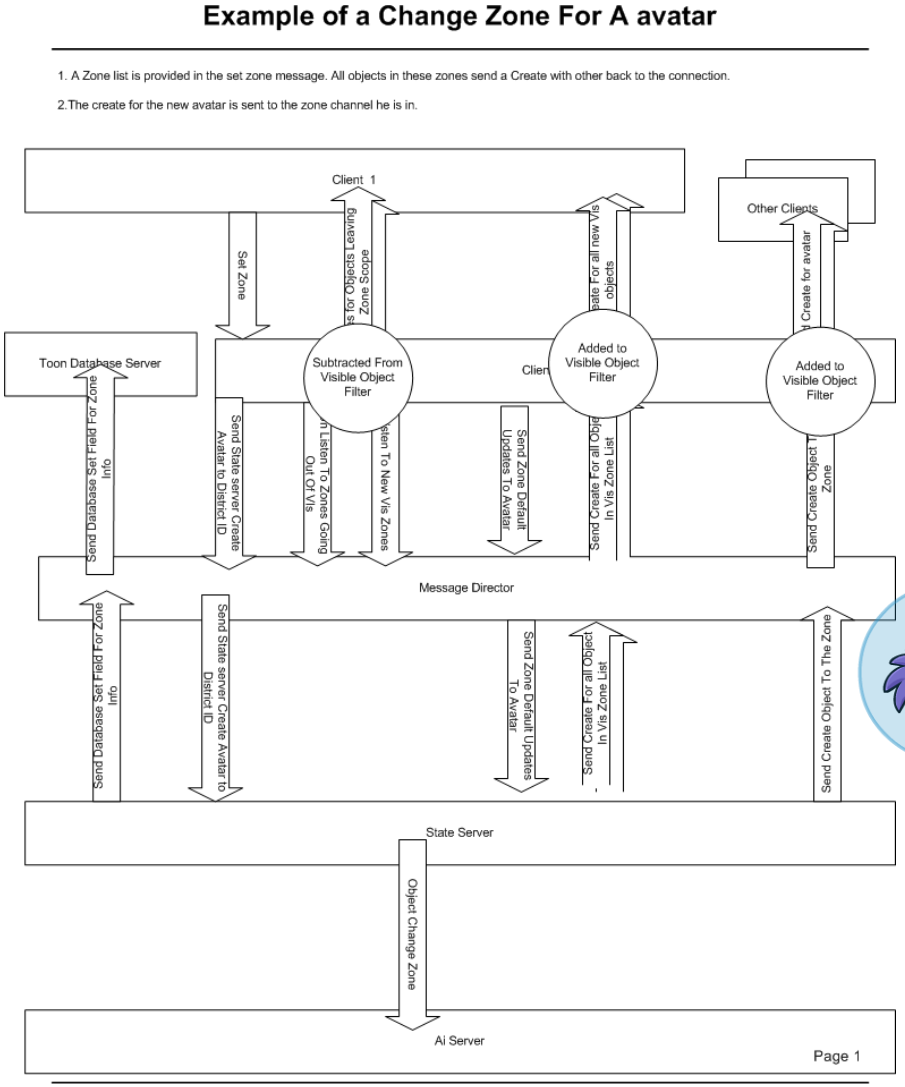
Channels + DC = Functional Cluster

- Channels.
  - Instance Channel.
  - Locations Channel.
  - Owners Channel.
  - Controlling AI Channel.
  - Persistent Store Channel.
- DC Class Definition.
  - How to read/write.
  - What to do with it.

Logical system



Asynchronous very finite units



What are DC Classes

- Avatars.
- Game object .
- A world.
- Every physical process.
- RPC type services.
- ….

- We like classes better than function dispatching.

What are Channels.

- Every distributed object has a channel.
- Every Account has a unique channel.
- Every guild, group…

- A channel is like a phone number. If you want to talk to it you use the channel.

A Game-Server Topology

- Distributed Class systems.
- Messaging Bus.
  - It's not a hierarchy it's a cloud.
- Network Culling.
  - Rooms/Zones.
  - Interest.
- Persistent store is not linked to update.
  - Asynchronous.
  - Memory image with a trickle writer.
  - Update merging.

General Rules for Server Systems

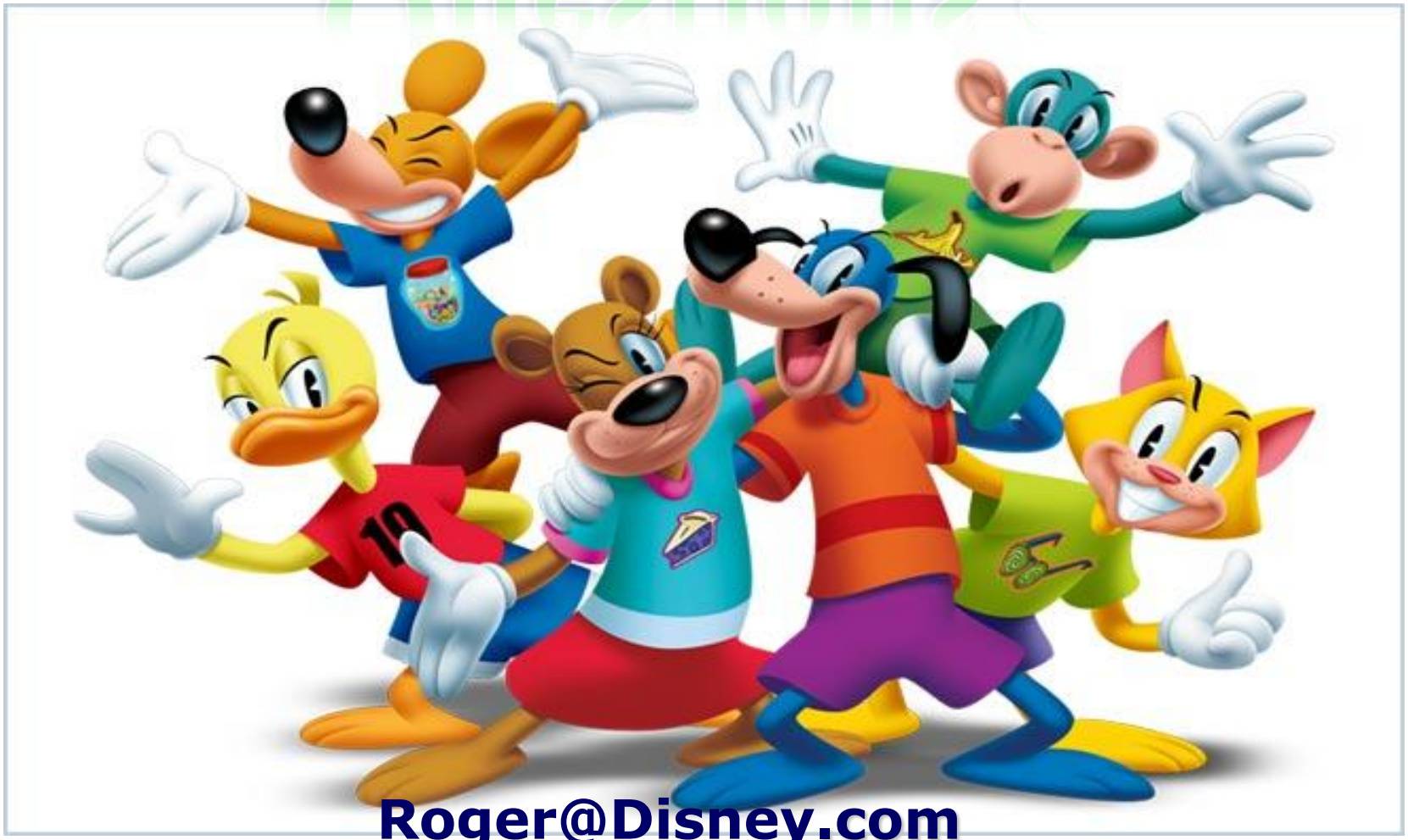
- Commodity before proprietary.
- Protocol over library.
- Asynchronous over synchronous.
- Process over threads.
- Horizontal over vertical scaling.
- Loose coupling over tight integration.
- Fail soft!!!!
- Optimizing game development life cycle is critical.
- Have a budget.
- Classes over functions.
- Keep it as simple as possible.

- Let the problem pick the technology!

War Stories (time allowing)

- You own it.
- Fake it.
- Know the real question.
- Hidden Races
- DBMS replication is for recovery, not performance.

Questions?



Roger@Disney.com