BIG PROJECT: PROJECT DARKSTAR

Karl Haberl, Seth Proctor, Tim Blackman, Jon Kaplan, Jennifer Kotzen

Sun Microsystems Laboratories

2008 Sun Labs Open House

Project Darkstar - Outline

- Some Background
  - The online games market, industry challenges, Darkstar goals, why Sun Labs?
- The Technology
  - Architecture, recent work, technical challenges
- Project Wonderland
  - A view of Darkstar from the developer's point of view
- Community
  - Current activities and plans for the future

Project Darkstar: Background

Online Games Market

- Divided into 3 groups:
  - Casual/Social – cards, chess, dice, community sites
  - Mass Market – driving, classic, arcade, simple
  - Hardcore – MMOG, FPS, RTS
- Online mobile still very small
- Online games are currently the fastest growing segment of the games industry
- Online game subscriptions estimated to hit $11B by 2011* **(Source: DFC intelligence)*
  - not including microtransactions, shared advertising, ...

The Canonical MMOG: World of Warcraft

- Approximately 9 million subscribers
  - Average subscription : $15/month
  - Average retention : two years +
  - $135 million per month/$1.62 Billion per year run rate
  - For one game (they have others)
- Unknown number of servers
- ~2,700 employees world wide
- Company is changing
  - Was a game company
  - Now a service company

World of Warcraft™ is a trademark and Blizzard Entertainment is a trademark or registered trademark of Blizzard Entertainment in the U.S. and/or other countries.

Ganz - Webkinz

- Approximately 5+ million subscribers
  - > Subscription comes with toy purchase
  - > Subscription lasts one year
  - > Average 100k users at any time
  - > Currently only US and Canada; soon to be world wide
  - > Aimed at the 8-12 demographic
    - And their mothers...
- The company is changing
  - > Was a toy company
  - > Becoming a game/social site company

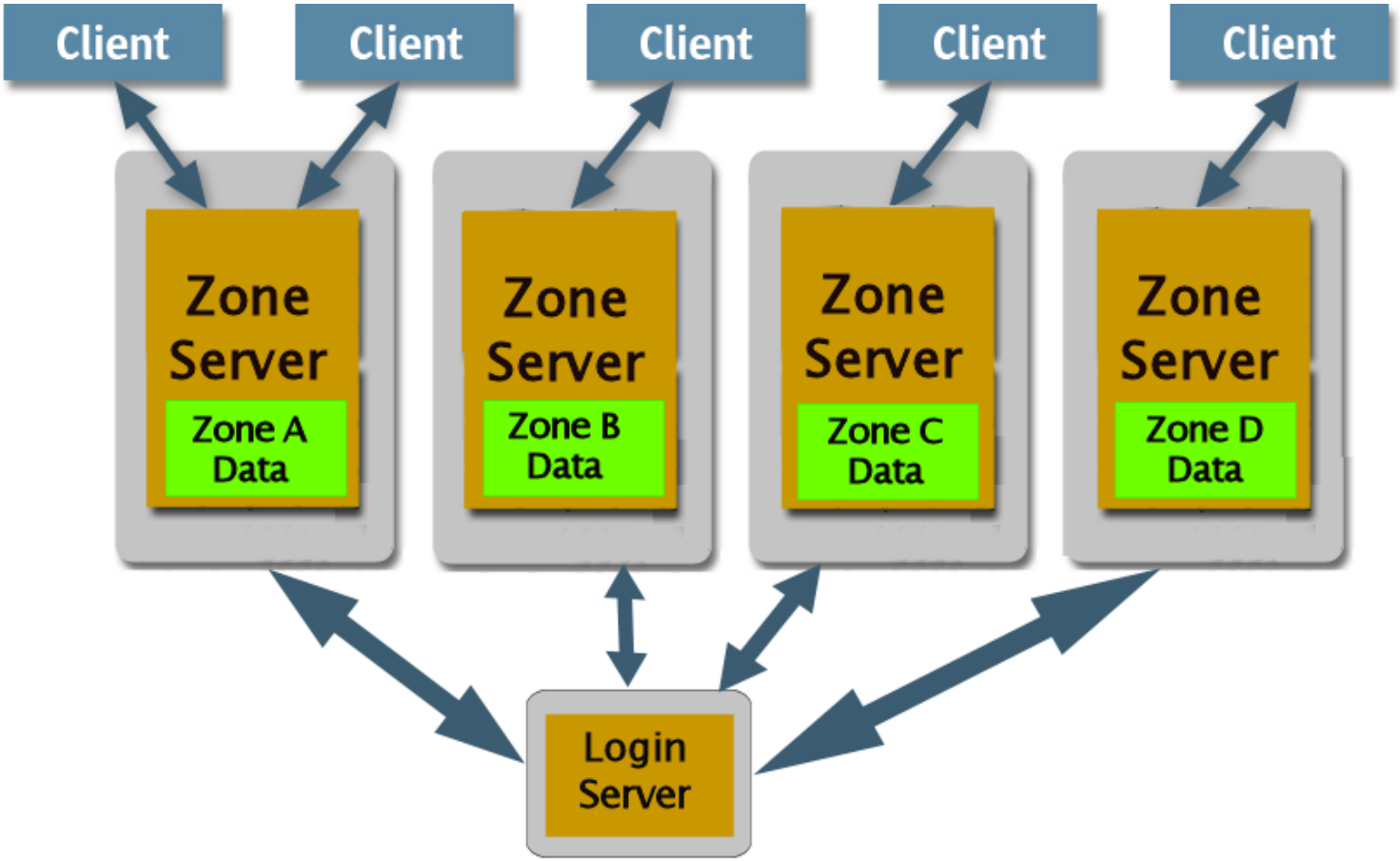
Habbo Hotel

- Virtual hotel for teens
  - > 89M accounts
  - > 8.3M unique users (12/07)
  - > 100K concurrent users peak
  - > will break 1B page hits per month
- Most revenue from content sales
- Grew to 1M users in first year
  - > started w 5 servers and 2 admins
- “Scaling was challenging” - Sulka Haro

Current Scaling Techniques

- Geographic Decomposition - “Shards”
  - One server = some geographic area
    - WoW: realm, Second Life: island, Nicktropolis/Webkinz: room
  - Need to decide scale during production
  - Get it wrong, game play impacted
  - When server is full, must connect to a different shard
  - No communication between shards; bad for guilds
  - Empty shards = idle servers, poor utilization
  - For social/casual, can be confusing for kids and adults

Sharded Architecture

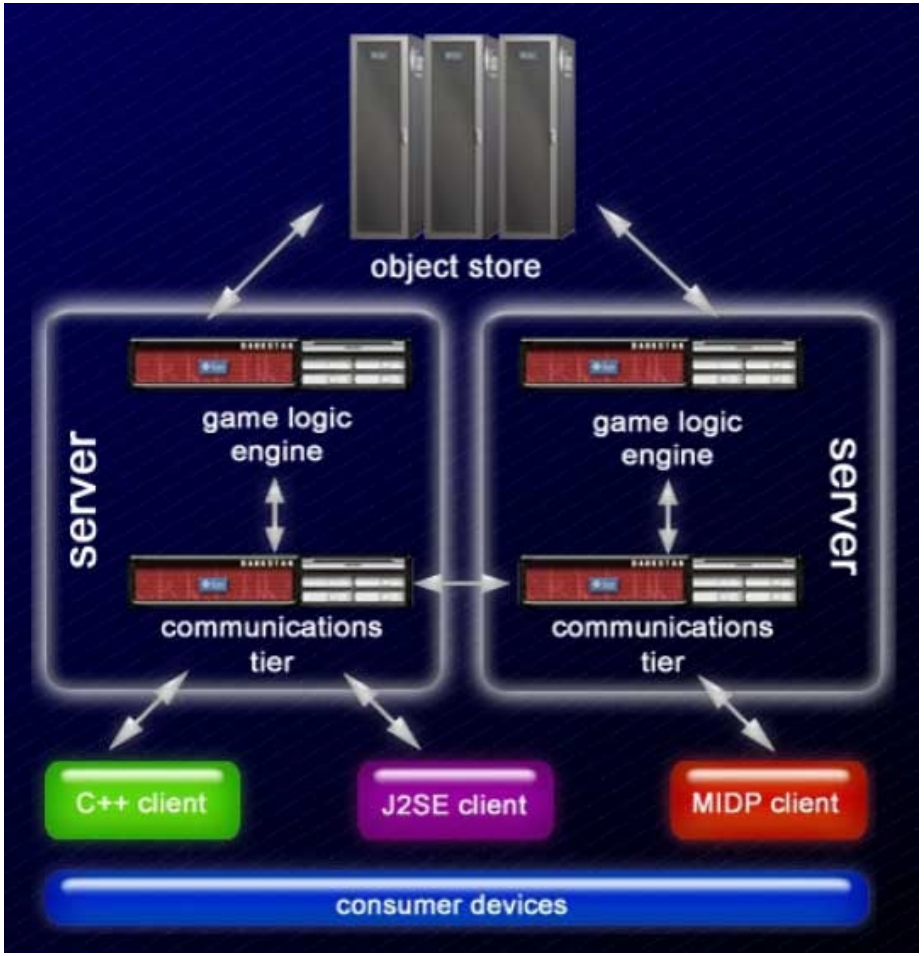


State-of-play for on-line games

- Difficult and expensive to develop, deploy, and manage
  - $30M + multi-year development for big-time MMOGs
  - Scaling requirements can vary wildly from projections
  - Very risky – hard to predict success of game in market
  - Only the big guys can play, limits innovation
- Scale and reliability are needed
  - Sharded architectures limit scalability and player interaction
  - Game developers are not networking or concurrent programming experts (nor do they want to be)
  - One call to customer service = ~3 month subscription
  - Chip architectures are changing – threads, not clocks!

Project Darkstar

- A software server designed to change the develop-and-deploy model for multiplayer online games and virtual worlds
- Written entirely in the Java™ programming language
- Game agnostic and platform agnostic
- Available as open source under GPLv2 license
  - > Commercial licenses and support can be provided
- Research project in Sun Labs



Project Darkstar Goals/Differentiators

- Enterprise class performance
- Simple programming model
- Shardless architecture
- Not a game engine
- Dynamic load balancing
- Server utilization
  - Higher efficiencies
  - Infrastructure flexibility and reuse
- Open and extensible
  - 100% Java technology
  - Open Source – GPL v2

The View from Sun Labs

- Interesting technology challenges and open questions – high risk
- A new potential market for Sun
- A different kind of research project
  - core technology, community and business models all being developed simultaneously
- Constant challenge to manage expectations while building community around an evolving, as-yet unproven technology

Project Darkstar: Core Technology

Outline

- Introduction to the Project Darkstar technology
- Changes for “multi-node”
- Current and future research

Project Darkstar Goals (1)

- Massive scale, low-latency platform
- Highly durable and fault-tolerant
- Coherent
- Familiar, event-driven model
- Easy to develop against

Project Darkstar Goals (2)

- Server-side model
  - Single system
  - Single-threaded
  - No concurrency complexity
    - Contention becomes the key issue
- Client-side model:
  - Very simple protocol
  - Platform-agnostic
- This talk focuses on the server-side

The Project Darkstar Stack

- A single server has 3 conceptual layers
  - Application code
  - Managers & Services
  - Core components
- Applications are developed...
  - In a transactional, event-driven model
  - Using Managers to access the system

Services and Transactions

- Services provide most core facilities
  - Data Service for shared data store
  - Channel Service for group communication
  - Task Service for scheduling durable events
  - Session Service for client sessions
- Services “see” the transaction model
- Extension API for writing custom Services

Core Components

- Basic facilities to support Services
  - Transaction coordination
  - Transactional scheduling
  - Non-transactional scheduling
  - Authentication
  - Profiling

Outline

- Introduction to the Project Darkstar technology
- Changes for “multi-node”
- Current and future research

Initial “Multi-Node” Goals

- A clustered solution
  - The stack runs on multiple machines
  - State is shared between these instances
  - Nodes can be added or removed
- An early 1.0 developer release
  - Applications work the same as in single-node
  - Focused on behavior, not performance

Initial “Multi-Node” Non-Goals

- Strong horizontal scaling
- Resource-aware load-balancing
- No single points of failure
  - More on these later in the talk

Concepts for Multi-Node (1)

- Shared data space
  - Replicate the view of the data store to all servers
  - Provide consistency across the cluster
- Node as an abstraction
  - Single Project Darkstar server instance
  - Application and core nodes

Concepts for Multi-Node (2)

- Node monitoring
  - Availability of nodes
  - Failure and recovery
- Identity mapping
  - All work is done for some identity
  - Each identity is assigned to a node
- Task migration

Updating the Data Service

- Embedded store exported to network
  - Data Service instances are clients
  - Objects kept at server
  - Contention managed at server
- Darkstar transactions mapped to data store model
  - Transaction time-out
  - Connection leasing

The Watchdog Service

- Supporting Service for node management
- Node resolution
  - Query for available nodes
  - Resolve state of a given node
  - Listen for node status changes
- Node recovery
  - Notification on node failure/shutdown
  - Verify recovery handling

The Node Mapping Service

- Supporting Service for identity mapping
  - Find where identities are mapped
  - Assign un-mapped identities
  - Resolve the identities on a given node
- Listen for mapping updates
- Manage identity status

Updating the Session/Task Services

- Session Service redirects connections
  - Handle moving sessions based on mappings
  - Vote status active on the connected nodes
- Task Service migrates tasks
  - Tasks are persisted in the data store
  - Tasks are handed-off as identities move
  - Tasks will only run once in the cluster

Core (Server) Node

- Standard Project Darkstar stack
  - No application code running
- Limited Services
  - Data, Watchdog, and NodeMapping
  - Services are run in “server mode”
  - Data is persisted at this point

Outline

- Introduction to the Project Darkstar technology
- Changes for “multi-node”
- Current and future research

What Was Successful in 1.0?

- Working prototype
- Use one or many nodes with same development model
- Community is increasingly active
- Learn about how to build out this environment

Where Does That Leave Us?

- Building a system meant we set aside some research
- Community wants quick fixes to hard scaling issues
- We have what we need to start investigating the hard research problems

Future Research Topics – Short-term and Long-term

- Throttling and push-back
- Improving concurrency
- Transactions and contention
- Scheduling
- ...

Throttling and Push-back

- Avoid system overload when there are too many requests
- Provide push-back to clients and other components

Throttling in the Current Release

- Delay reading next session message until current message task is done
- Throw `MessageRejectedException` when:
  - Sending session message if session write buffer is full
  - Sending channel message if local buffer is full
    - May still overwhelm other nodes due to fanout

Throttling: Future Work

- User login
- Reading session messages when task queue is full
- Scheduling new tasks when task queue is full
- Sending channel messages when node send queues are full
- Feedback for other communication between components

Improving Concurrency

- Applications need to be designed to avoid concurrency hot-spots
- System should provide high-concurrency data structures
- System facilities should support high concurrency

Concurrent Collections

`ScalableHashMap` and `ScalableHashSet`

- Added in release 0.9.5
- Uses a TRIE structure to improve concurrency
- For applications and services

Concurrent Queue

- Provide read/write concurrency
- For applications, also client session and channel services
- Provide write concurrency using “funny” semantics
  - Extra concurrency possible if pending added elements cannot be read
  - For services only

Data Service Object Allocation

- Concurrent allocation in current release
- Problems with page-level locking
- Problems with object-level locking
- Use random allocation
- Support explicit object clustering

Observing Contention

- Contention is a key performance issue
- Core scheduler runs all tasks
  - We know when transactions conflict
  - We don't know much about the cause
- Currently adding:
  - Tracking for what objects are being accessed
  - What transactions are causing conflict
- Very useful for debugging/profiling

Reacting to Contention

- Drive re-try policy
- Exploring full contention coordinator
  - Define our own policy for priority
  - Dictate optimism/pessimism for locking
  - Requires coordination between nodes
- Provide feedback to the scheduler

Transaction-Aware Scheduling

- Project Darkstar scheduling requirements
  - Low-latency
  - Low-jitter
  - Whole task execution
- Scheduler is aware of transactions and results
  - Handles re-trying aborted transactions
- Scheduler is aware of task operations

Node-Local Scheduling

- Fair priority-scheduling...
  - ...for some definition of “fair”
- Task-predictive scheduling
- Currently exploring:
  - Better policies for re-try on transaction abort
  - Using a contention coordinator to track patterns
  - How transaction coordinator and scheduler can better collaborate

Scheduling for Multi-Node

- Node-level scheduling may be affected by...
  - Contention with other nodes
  - Re-try for transactions across nodes
  - ...easier when contention is localized
- Node-level scheduling should feed-back...
  - What objects are needed?
  - Who is interacting with whom?
  - How well is the node keeping up with load?
- Exploring how to work with mapping and caching for more efficient cluster-wide scheduling

Other Research Topics

- Caching and partitioning in the data store
- Load balancing
- Zones/shards versus a shardless world
- Persistence and backing stores
- External transaction coordination
- Monitoring and management
- Still others that we can explore now that we have a working prototype...

Wonderland: A Project Darkstar Application

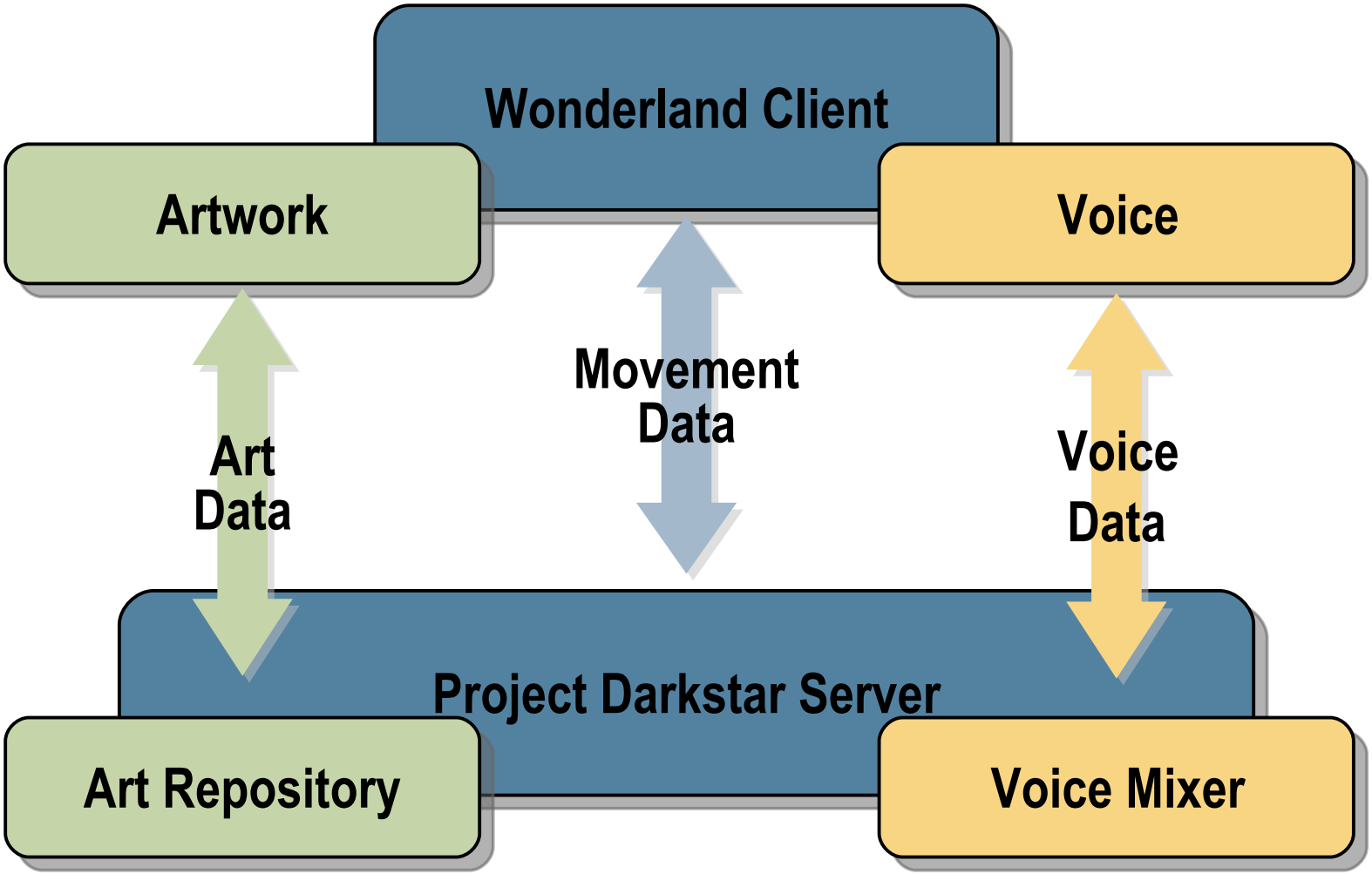
Project Wonderland

- Business grade virtual world toolkit
- Open source, Java-based, highly extensible
- Implemented as serious game on top of Project Darkstar

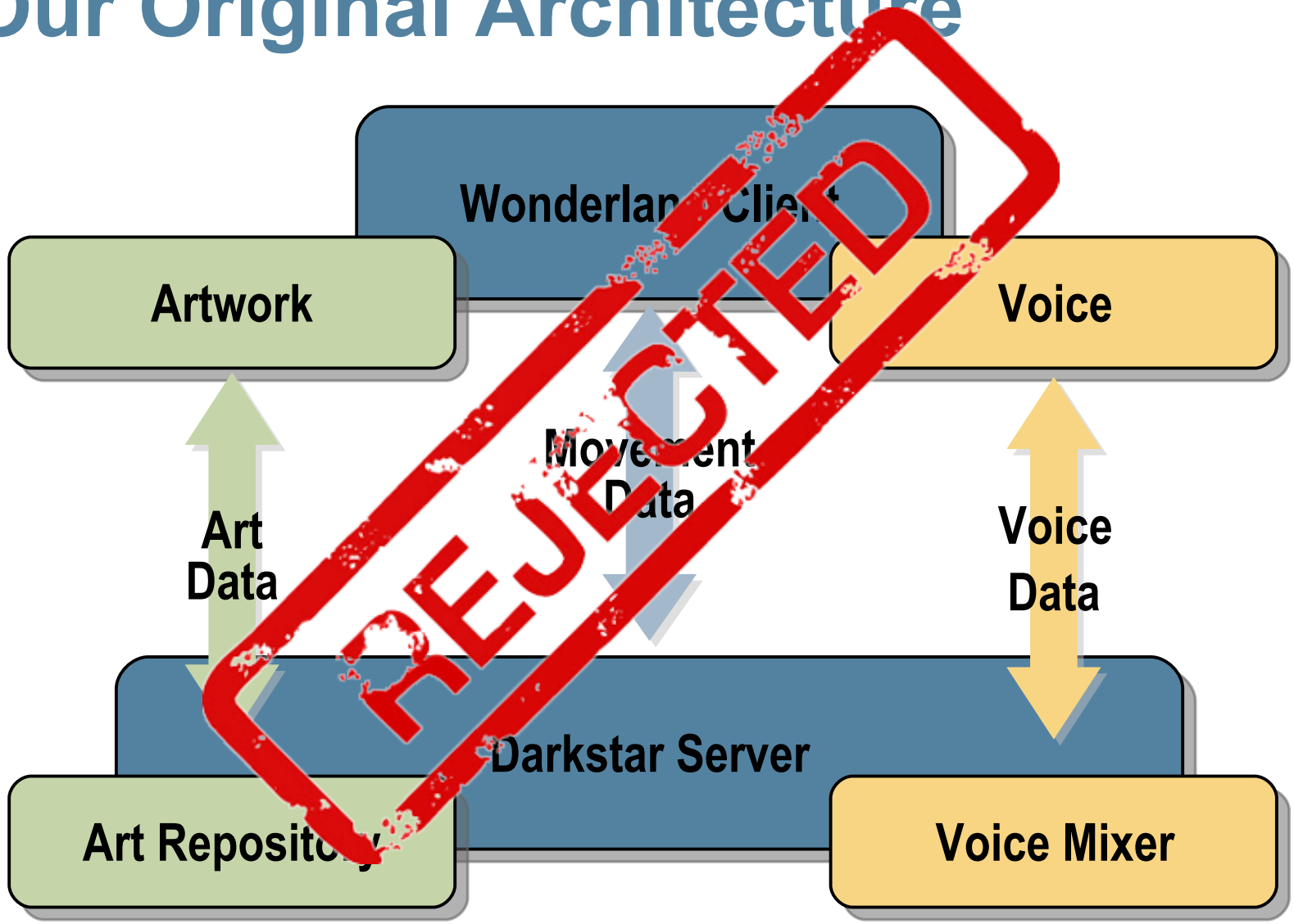

Demo Video: MPK20
http://wonderland.dev.java.net

Lesson 1: Communications

Our Original Architecture



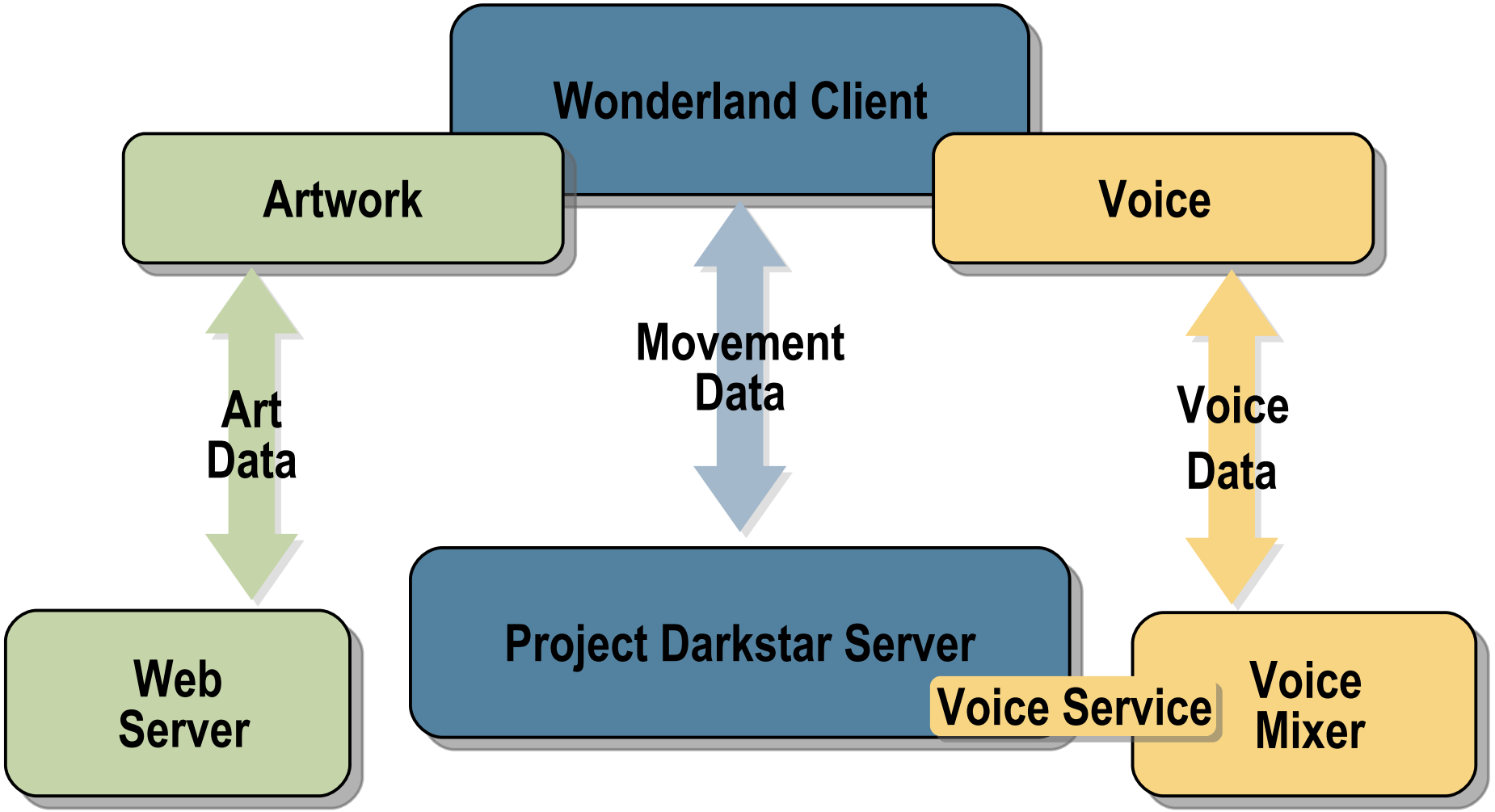
Our Original Architecture



Project Darkstar Channels

- Problems:
  - Buffer overruns
  - Timeouts
  - Poor throughput
- Designed for:
  - Coordination & control
  - Small messages
  - High throughput
- Bad ideas:
  - Bulk data transfer
  - Continuous streams of data

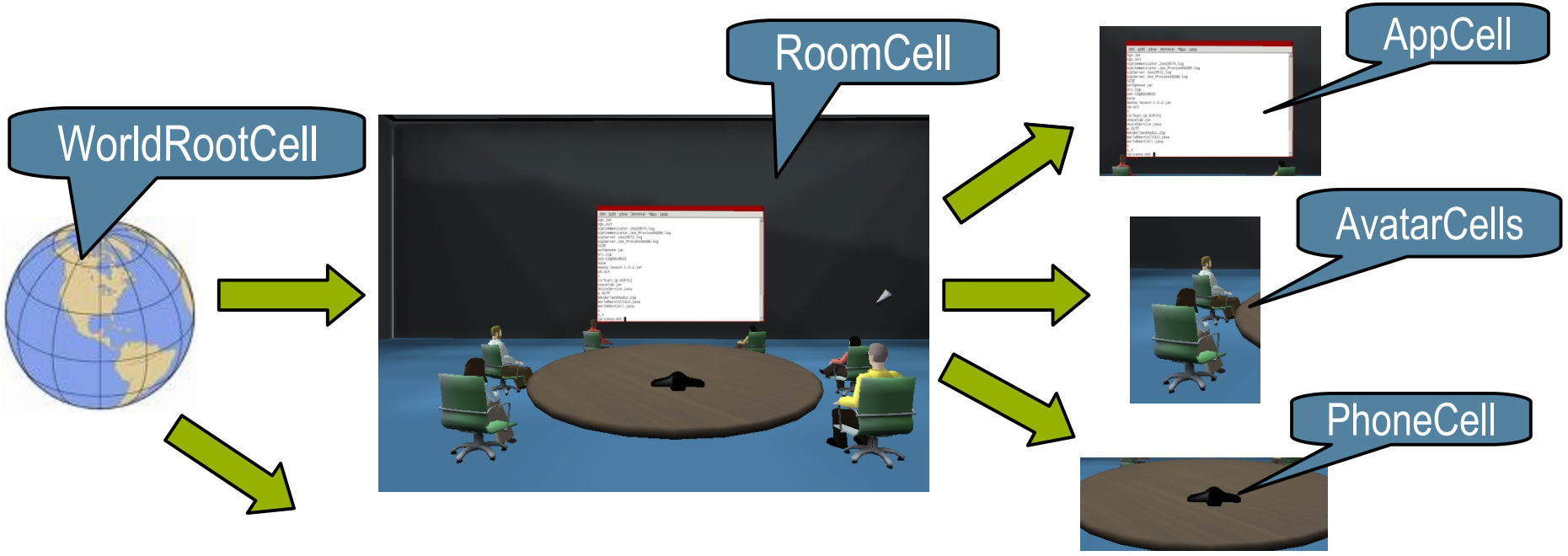
Using Other Transports for Bulk Data



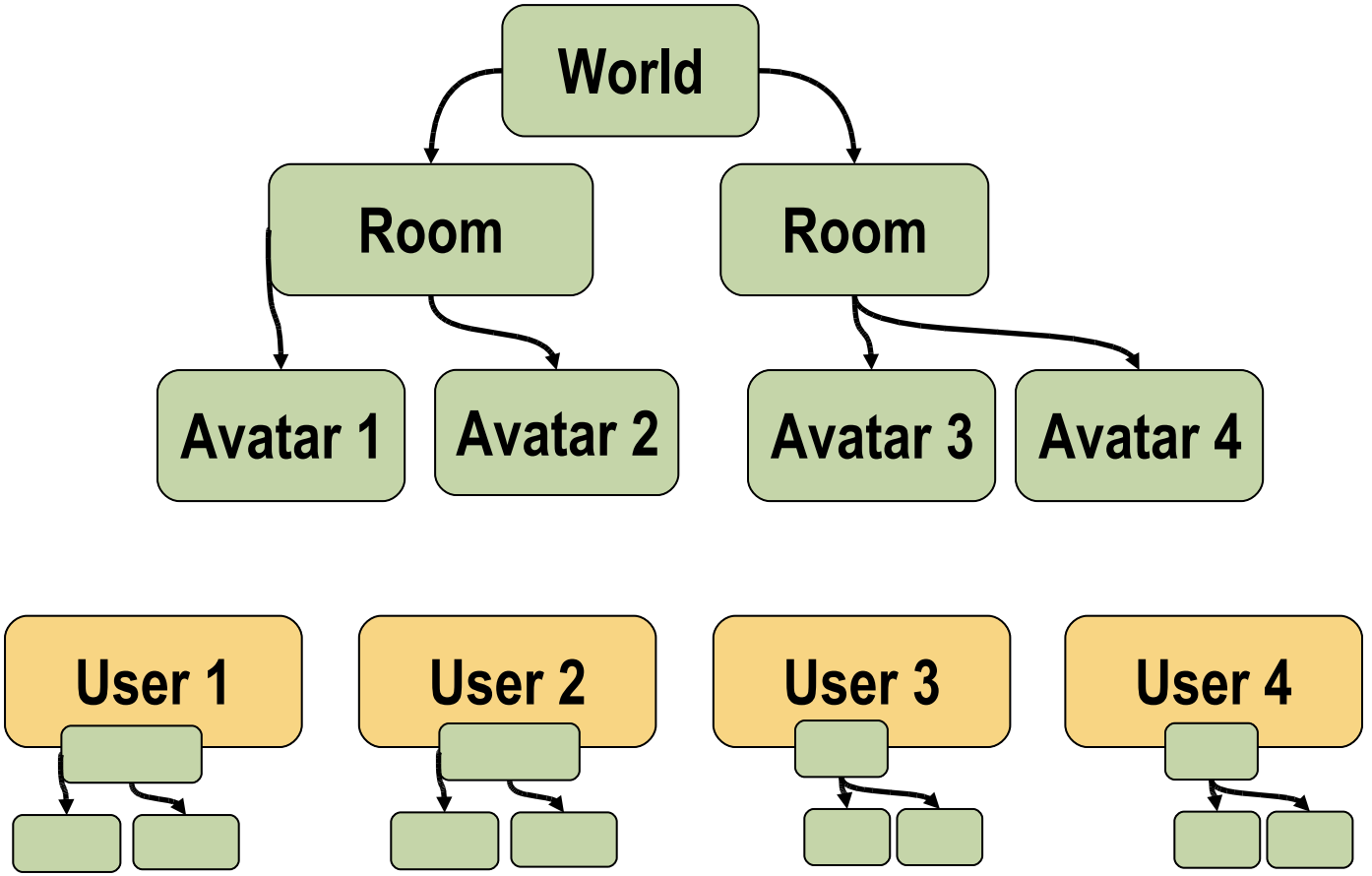
Lesson 2: Data Store Access

The Cell Hierarchy

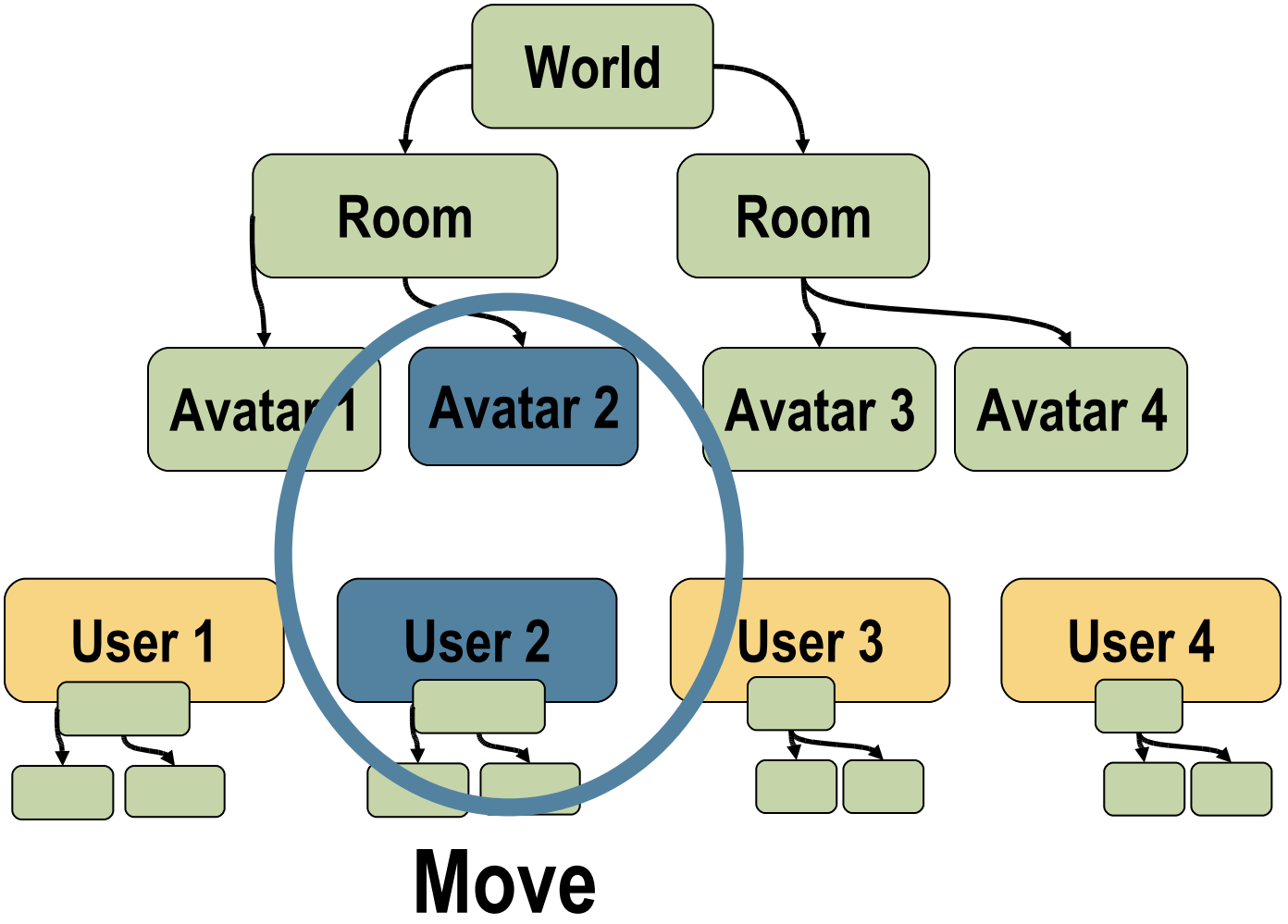
- World is divided into discreet volumes called “cells”
  - > Cells are nested into a tree structure
  - > Each cell is a Project Darkstar ManagedObject
  - > User MOs maintain a list of cells they can see



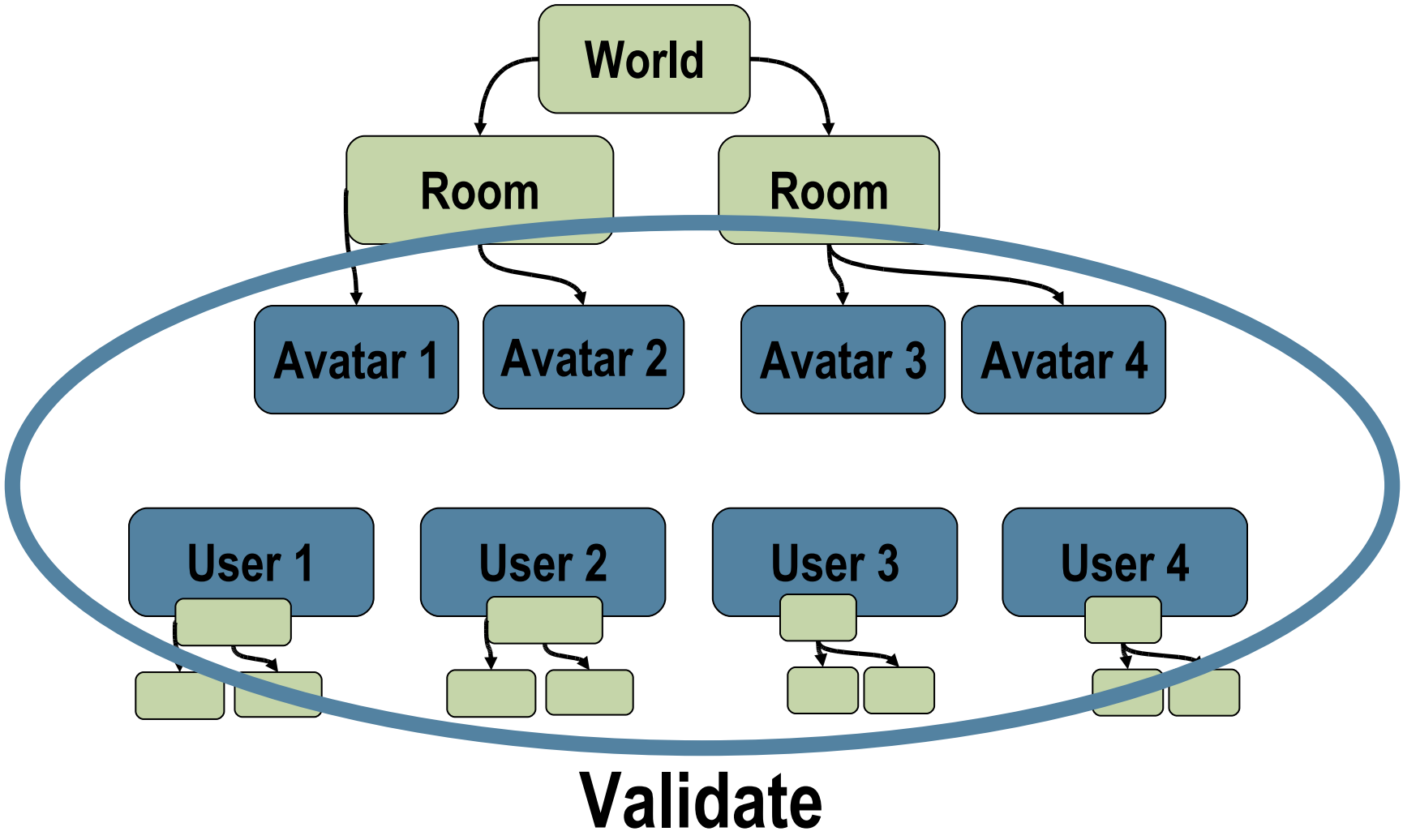
Original Move



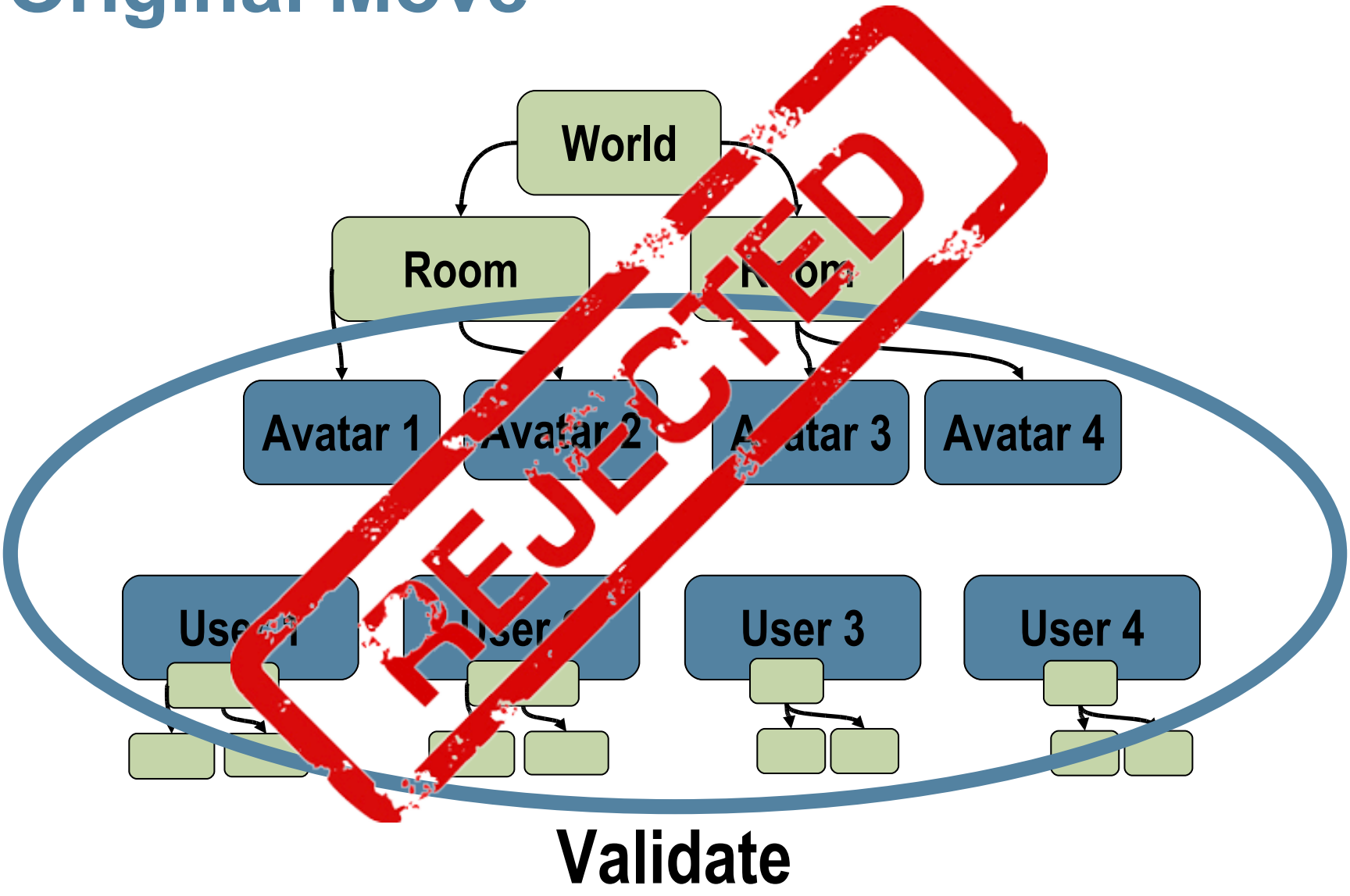
Original Move



Original Move



Original Move



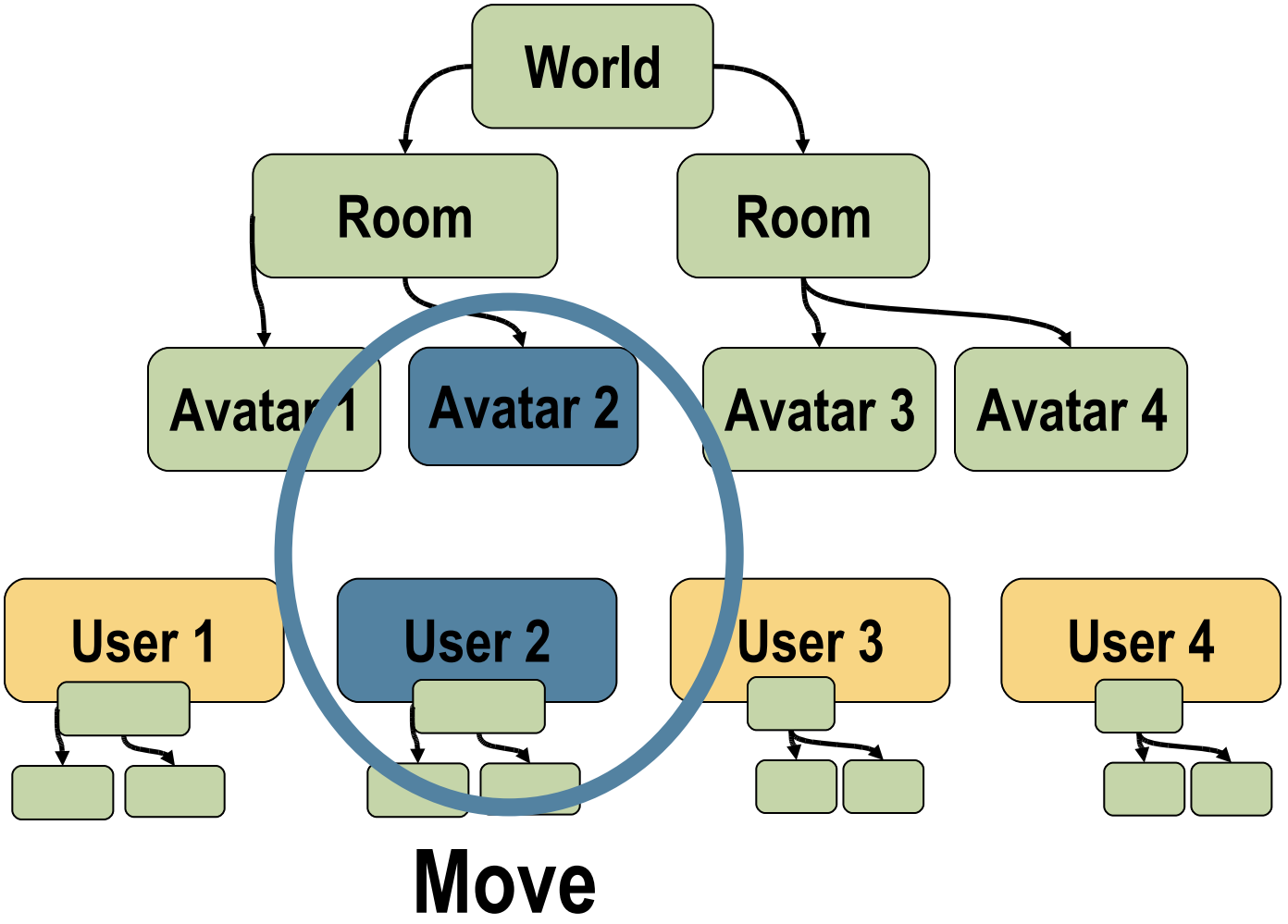
Project Darkstar Data Store

- Problems
  - Timeout
  - Conflict

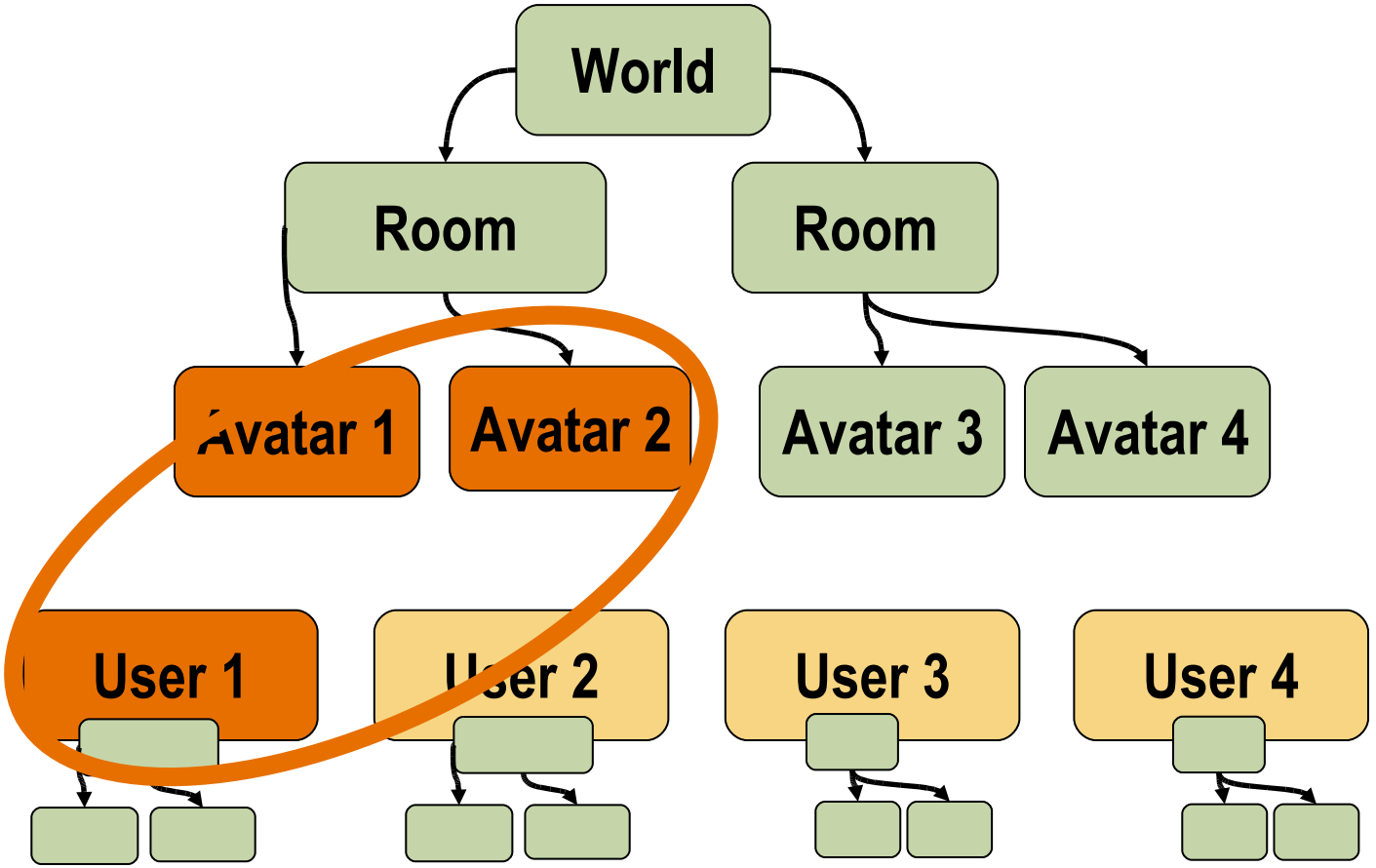
- **Designed for:**
  - **Small objects**
  - **Frequent access**
  - **Transactions**
  - **Lots of writes are OK**

- Bad ideas:
  - Access too many objects in one transaction
  - Store huge objects

Keep Transactions Small

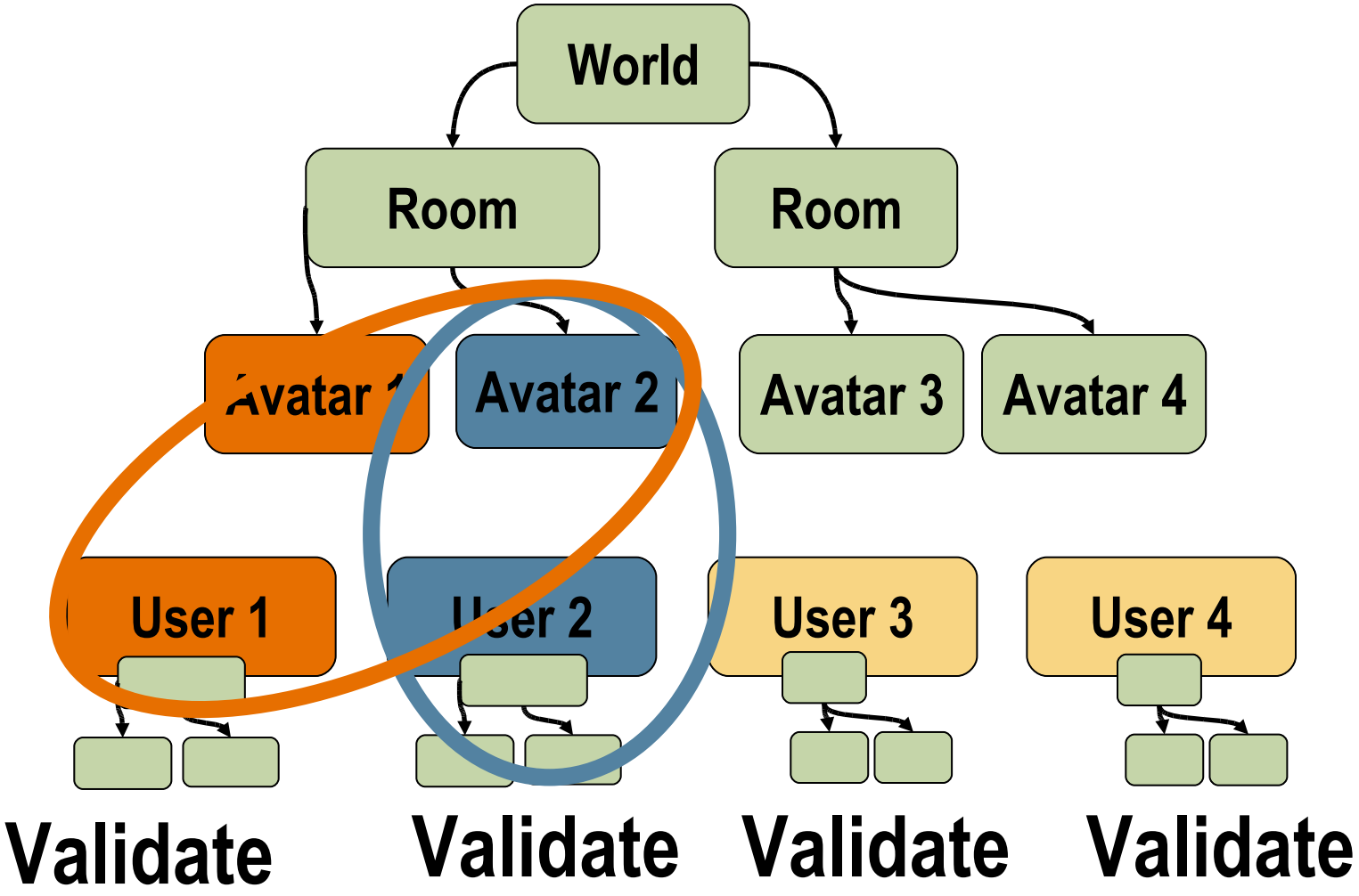


Keep Transactions Small

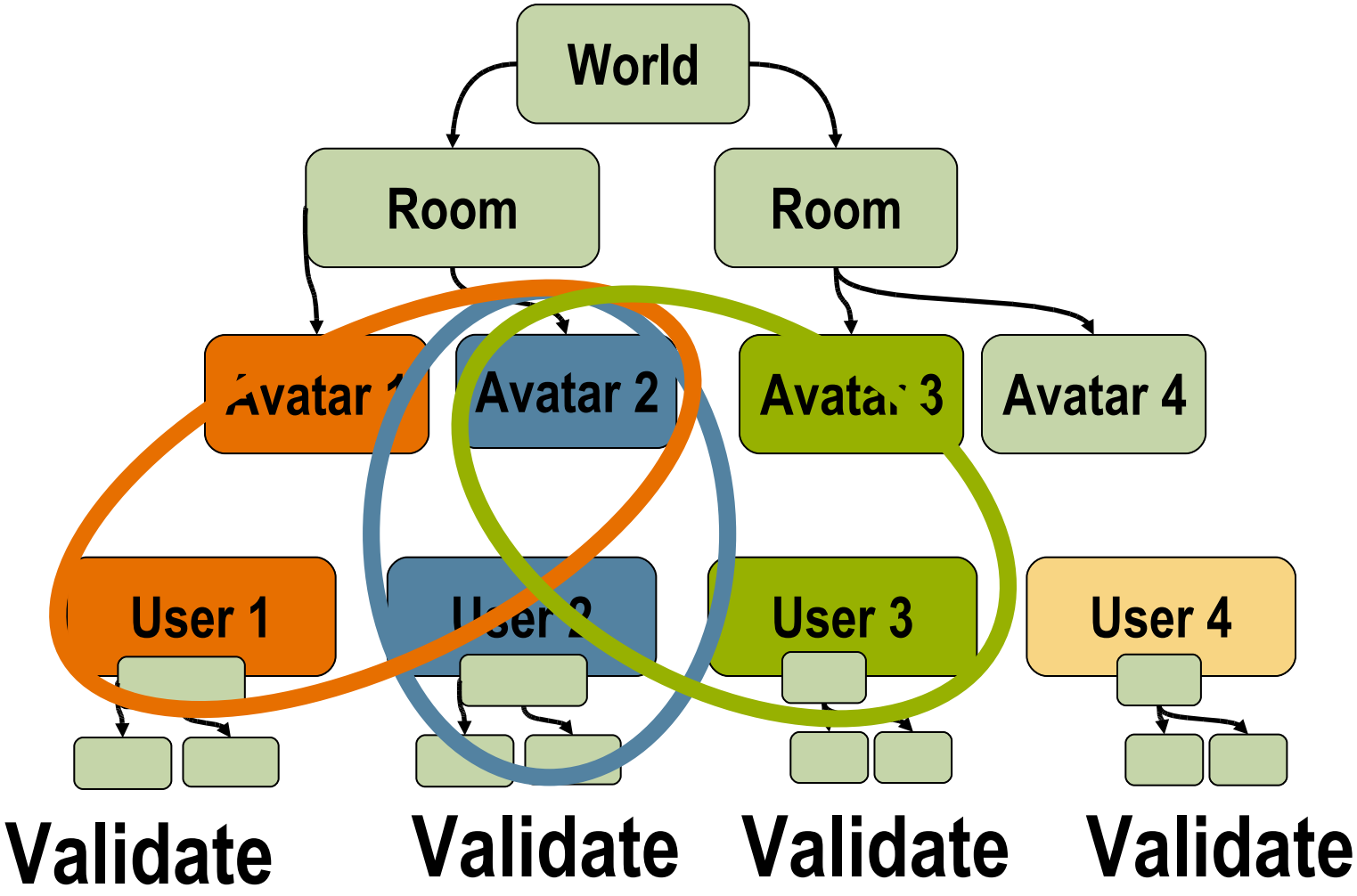


Validate

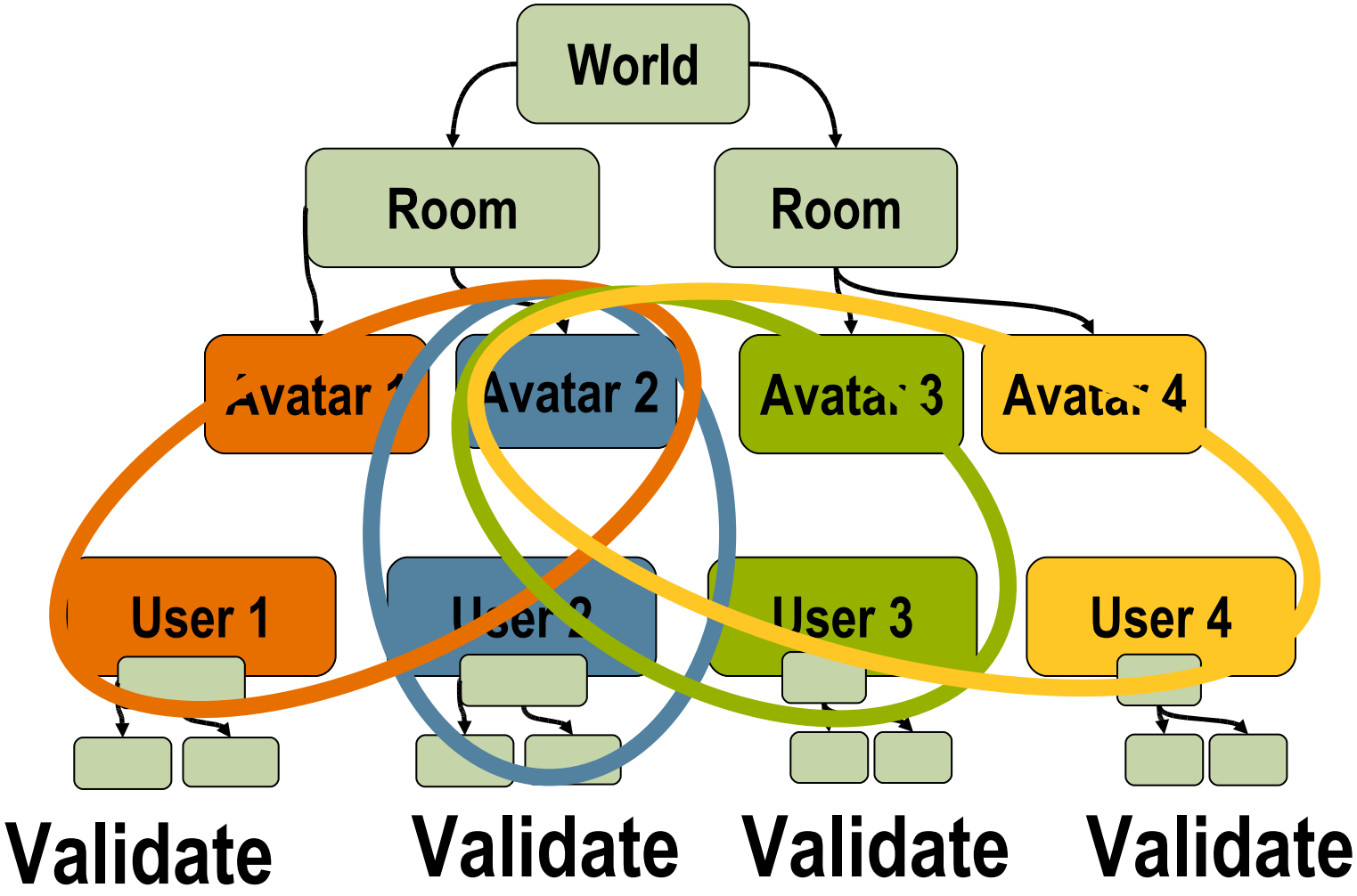
Keep Transactions Small



Keep Transactions Small

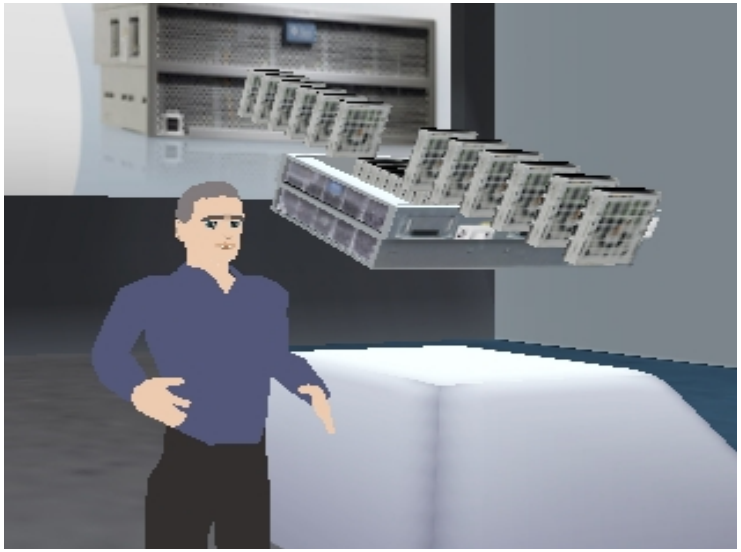


Keep Transactions Small



Project Wonderland Resources

- wonderland.dev.java.net
  - Download today!
  - Community
  - Forum
- Visit our demo
- blogs.sun.com/wonderland



Project Darkstar: The Community

Who or What is a Community?

- Fellowship of people
- Common attitudes, interests, goals
- Shared social values and responsibilities
- Practicing common ownership

Why Community?

- Model has proven effective for advancing tech
- Can a community achieve a broader goal of technology adoption?
  - Core tech development
  - Documentation
  - Add-on and complementary tech dev & integration
  - Tech support
  - Education and consulting services
  - Hosting services
  - Promotion
  - Distribution

The Project Darkstar Community

- Common interest: Project Darkstar technology
- Common goal: community health and growth
- People: all those working with the technology
  - Roles: users, developers, service providers, advocates, others
  - Entities: individuals, companies, organizations
  - Interests: personal, commercial, private, public
  - Cultures: online games/entertainment, large scale systems/business
  - Engagement: users, contributors, leaders
  - Geographies: global

Shared Social Values & Responsibilities

- Free and open source core
- Inclusive
- Transparent
- Contribution
  - The act
  - The artifact
  - The contributor
- Fun!

Enabling/Ensuring Shared Ownership

- Free and open source licensing
- Enable engagement
  - Members want to be involved; let them be involved
- Sun as a community member
  - Individuals and company
  - Leadership by merit, not entitlement
- Governance?
  - Only if/when needed

Questions?

Karl Haberl, Seth Proctor, Tim Blackman, Jon Kaplan, Jennifer Kotzen

[first.last@sun.com]