FORTERRA
SYSTEMS INC

Live Entity State Stream

Virtual World Interoperability: Inclusive integration across technologies

Jon Watte
CTO, Forterra Systems
jwatte@forterrainc.com

Learning From Experience

- Interoperability Experience from the following VW integration cases:
  - “Live, Virtual, Constructive”
  - Integrated System AAR
  - "NPC" Simulation
  - External Simulations (medical, traffic, ...)
  - Real-world Geometry (cities, oil rigs, ...)
  - External Analysis (performance, events, ...)
- Currently, those are solved using either task-specific or proprietary protocols.

Why solve simulation integration?

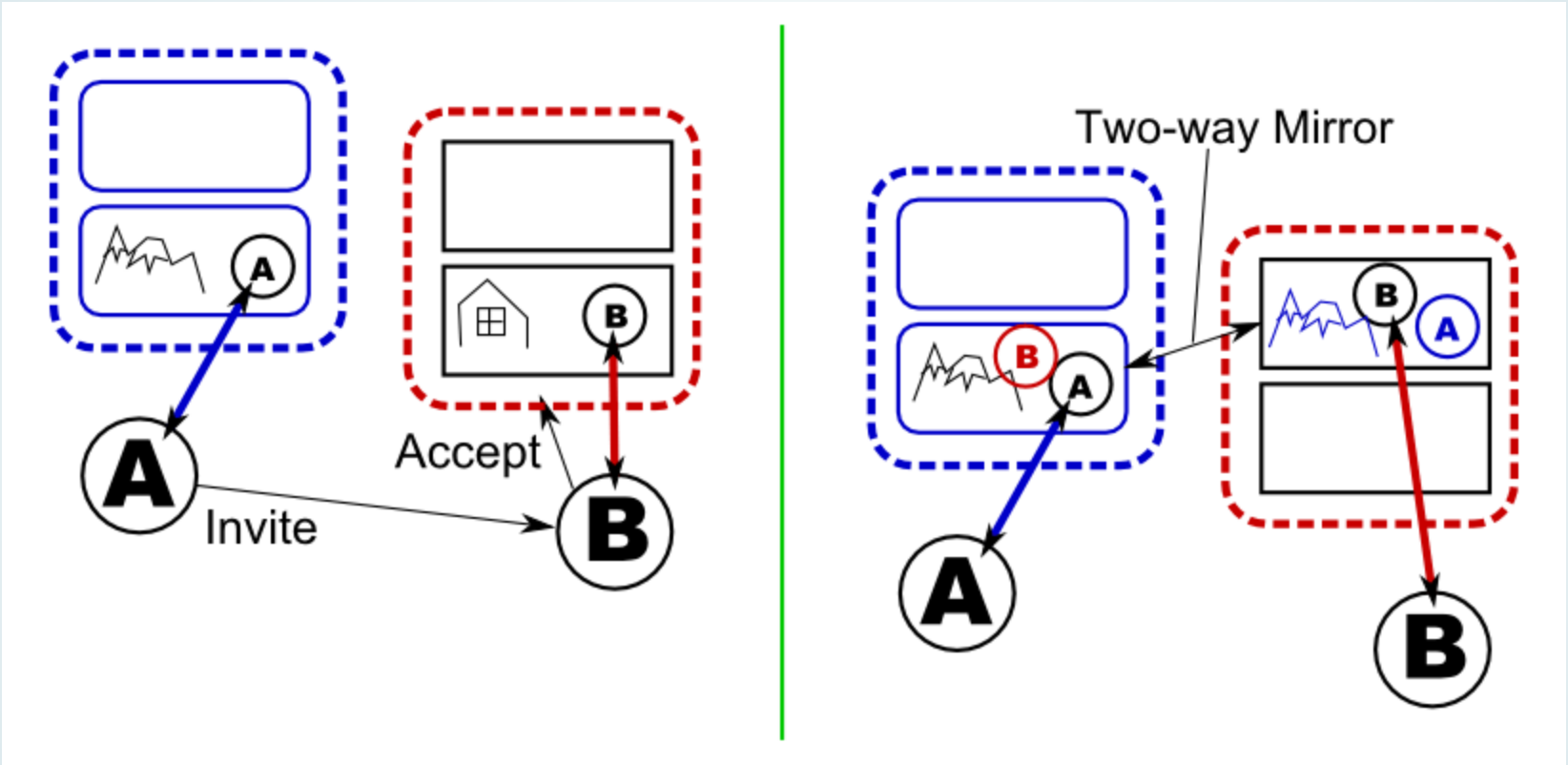
- "Teleport" isn't that interesting.
  You want to bring people *and environments* together!
- Moving live objects across technology boundaries:
  Technically hard and politically impossible!
- Compare 2D web: Mash-ups are hard because there are many server-side technologies (J2EE, ASP.NET, LAMP, etc)
- Virtual worlds are *real-time*, interactive simulations.
- The benefit of virtual worlds compared to 2D is *synchronous interaction*.
- Virtual world mash-ups need a protocol similar to SOAP or XML-RPC, but optimized for live, synchronous interaction.

The interaction model

- Someone on system A invites someone on system B through a URL.
- User on system B accepts invite. System B sets aside some space for connecting to system A, and puts user B there.
- System A provides environmental information to system B.
- A and B provide entity presentation to each other. This allows interaction.
- Objects homed on A stay executing on A; objects homed on B stay executing on B.
- This can scale to as many interacting systems as you want, because *simulation is not centralized*.
- Investment for each system provider is low, because the *existing client/server stacks remain unchanged*.

FORTERRA

Illustration (simplified)



Protocol Design

- Known property semantics ("Position," "DisplayMesh," etc).
- Publish/Subscribe on per-type basis, saves bandwidth.
- Schema allows mapping of extended objects to known kinds.
- Simple "interaction" and "tweak" RPC mechanism.
- Well-defined semantics for common interactions like "collide" or "activate" or "damage."
- Peers send presentation, not internals.
- UDP, TCP and HTTPS/Upgrade versions.
- Biggest bang for the buck implementation.
- Proven existing methods, synthesized protocol.

FORTERRA

Actions

- Define target use cases.
- Define the called-out whitespace.
  - Property semantics.
  - Interactions (collide, activate, etc).
  - URL formats.
  - Standard file formats (meshes, textures, animations, etc).
- Achieve an open sample implementation.
- Demonstrate that interop solves the required use cases.
- Document the learning into an RFC.

FORTERRA

FORTERRA

SYSTEMS INC