



# An Ecosystem for Interactive Mixed-Reality Applications on the Web

Tutorial on Mixed Reality for the Web  
Web3D '14, Vancouver

August 10th 2014

# Tutorial Outline



## Today's Topics:

1. The Future Internet Program
2. [Declarative 3D for the Web](#): XML3D and XML3D Assets
3. [Sharing one World](#): Multi Client Applications with XML3D
4. [Mixing the Real with the Virtual](#): Augmented Reality and the Reality Mixer
5. [Real Virtual Interaction](#)

# Tutorial Outline



## Speakers:

- [Kenny Mitchell](#) (Disney Research Zürich)
- [Marcel Lancelle](#) (ETH Zürich)
- [Fabio Zünd](#) (ETH Zürich)
- [Torsten Spieldenner](#) (DFKI Saarbrücken)

# Future Internet Program



## The Future Internet Program of the EU (FI-PPP)

- Large Integration Activity of the EU (500M Eur.)
  - Move from TCP/IP to **service-oriented architecture**
  - Create **comprehensive and consistent** set of services
- **5-year** duration (2011 - 2016)
  - Provide Core Technology in FI-Ware **Generic Enablers**
  - Create use-case scenarios in FI-Content (**Specific Enablers**)
- Build a **business ecosystem** around the technology
  - New approach: Include **SMEs and Web Entrepreneurs**

# Future Internet Program



## How can YOU use this 3D-Internet Technology?

- Check out **FI-WARE Generic Enabler**
  - XML3D source is already freely available on GitHub (<http://github.com/xml3d/xml3d.js>)
  - Other GEs will be available on **FI-LAB** (<http://lab.fi-ware.org>)
- Check out **FI-Content Specific Enablers**
  - Will be freely available at <http://mediafi.org>
  - SEs will be freely accessible on FI-LAB

# Declarative 3D: XML3D



## Declarative 3D for the Web:

- Extension to [HTML5](#) for 3D Content
- Entirely based on Web technologies: [DOM](#), [CSS](#), [HTML Events](#) etc.
- Generic data model that allows [data compositing](#) and external references
- [Xflow](#): Efficient declarative dataflow processing
  - Animations, image processing, AR, ...
- Provided as [polyfill implementations](#)



# Declarative 3D: XML3D



## Some more cool features:

- **Instancing mechanism** for externally described assets
- Efficient binary transmission format: **BLAST**
- **shade.js**: Portable and adaptable material description in JavaScript
- Integration of many external services:
  - **DFKI**: Scene editing, virtual characters, motion synthesis, server-based rendering, ...
  - **FI-PPP**: Synchronization, Augmented Reality, Real Virtual Interaction

→ Huge eco system for compelling 3D Web Applications

# Declarative 3D: XML3D



## Some more cool features:

- **Instancing mechanism** for externally described assets
- Efficient binary transmission format: **BLAST**
- **shade.js**: Portable and adaptable material description in JavaScript
- Integration of many external services:
  - **DFKI**: Scene editing, virtual characters, motion synthesis, server-based rendering, ...
  - **FI-PPP: Synchronization, Augmented Reality, Real Virtual Interaction**

→ Huge eco system for compelling 3D Web Applications



# Instantiating XML3D Geometry



## Instantiating assets in XML3D

```
1 <asset id = "myasset" transform="#baseTransform">
2   <assetmesh shader="shaders.xml#tex" src="cube.json" />
3   <assetmesh shader="shaders.xml#tex2" src="part_2.json" />
4 </asset>
```

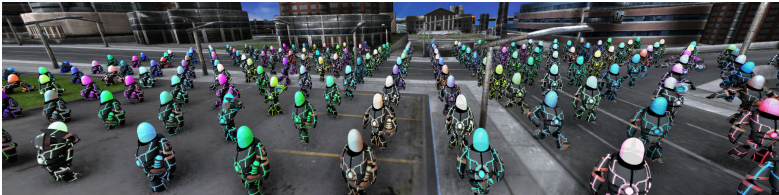
Instantiate assets with just one node:

```
1 <model id="instance_1" src="resources.xml#myasset"/>
2 <model id="instance_2" src="resources.xml#myasset"/>
```

## Assets also provide:

- Configurable parameters, e.g. for individual animations and poses
- *Can be defined externally!*

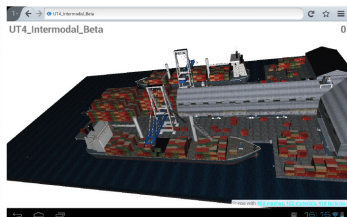
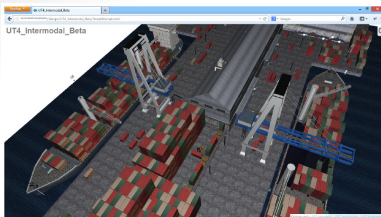
# Instancing XML3D Geometry



# XML3D Asset Servers



## XML3D - Repo



- Fusion of [XML3D](#) and 3D-Repo
- Provides [REST](#) Interface to request versioned 3D models
- Uses [BLAST](#) for transmission of large structured data  
*See also Web3D paper talk*

# XML3D Asset Servers



## ATLAS

*Advanced Three-dimensional Large-scale Asset Server*

Provides assets for different front-ends:

- Upload asset in supported format (e.g. *COLLADA*)
- Server stores asset in **internal interchange format**
- Front-end requests asset in specific format via URL
- Server converts asset and delivers it to front-end

Next:

- *Shared interactive 3D worlds based on XML3D*