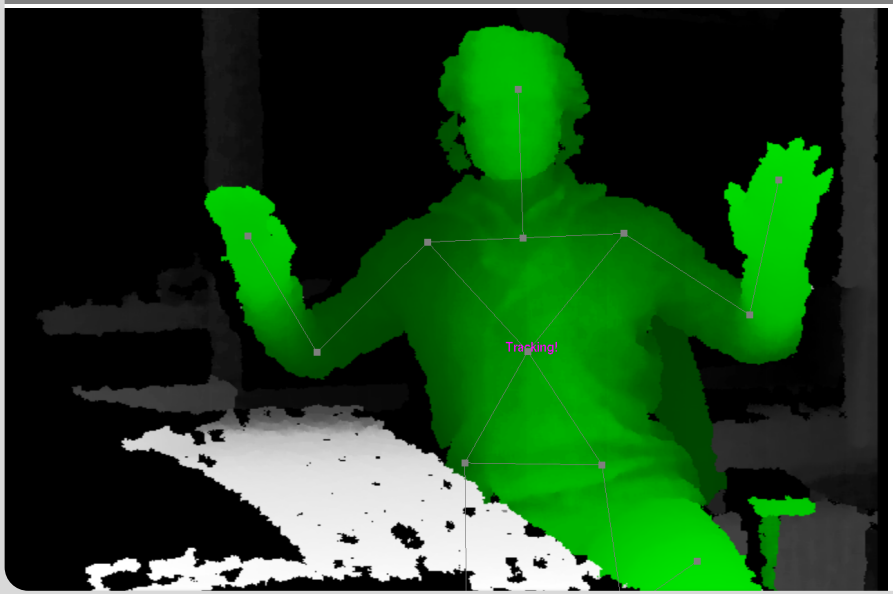
Karlsruhe Institute of Technology

Semantic RESTful APIs for Dynamic Data Sources

Felix Leif Keppmann / Steffen Stadtmüller

Institute of Applied Informatics and Formal Description Methods (AIFB)



Outline

Motivation

Proof of Concept

Approach

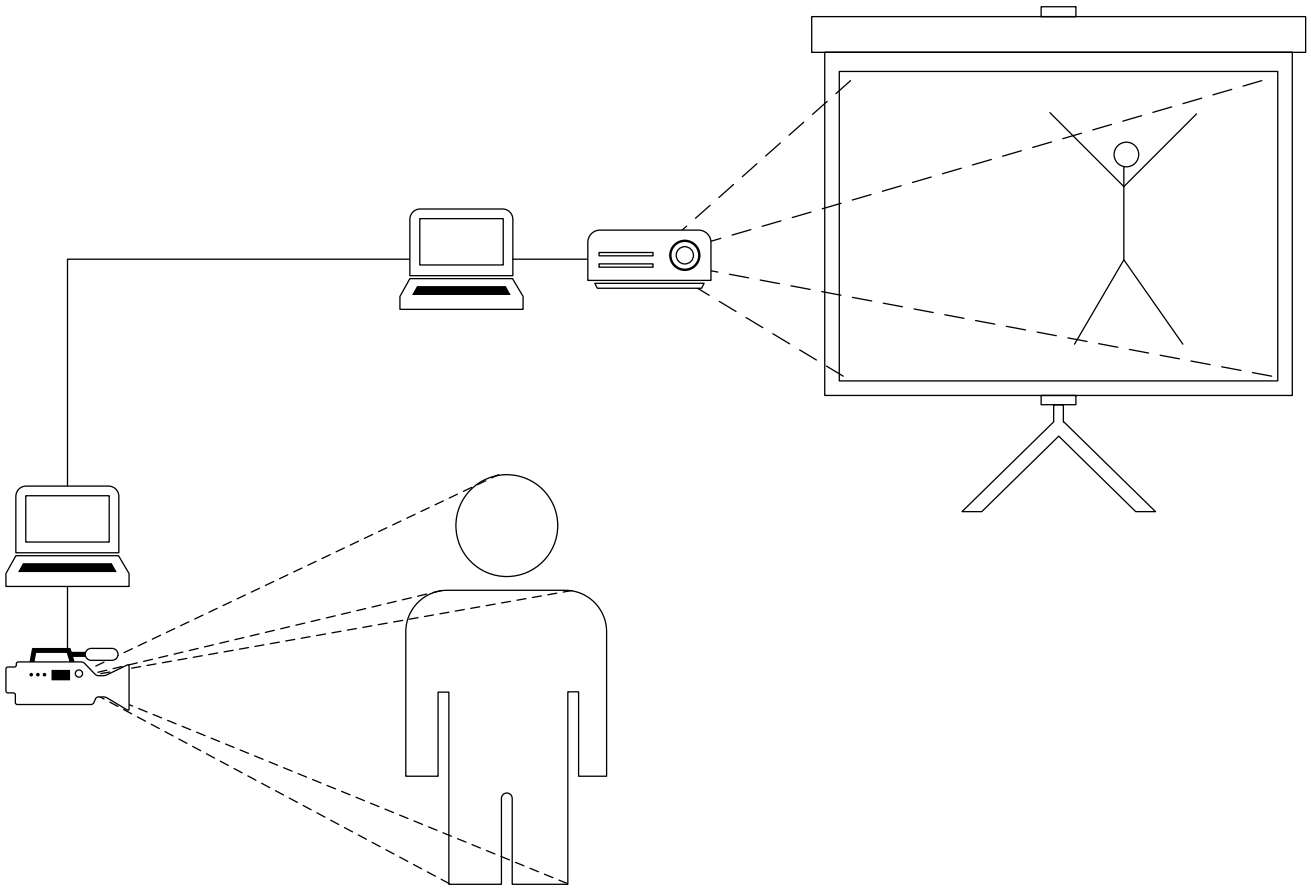
Evaluation

Conclusion

Outline

Motivation

Scenario



Motivation

Scenario

- Data sources
  - Video and depth sensors
  - High frequent updates
- Data sinks
  - Middleware, e.g. tracking or gesture recognition
  - Applications, e.g. visualization or augmented reality
- Issues
  - Several players in the community/market
  - Proprietary software stacks
  - Integration of devices and application
  - ...

Motivation

Proposal

- Exposing highly dynamic data sources via a semantic RESTful interface
  - Characteristics of a RESTful implementation
  - Combination with the dynamic nature of the original sources
  - Direct integration enabled via Linked Data
- Advantages
  - Data Preparation
  - Data Selection
  - Interoperability
  - Performance Decoupling
- Streams and REST design seen as incompatible

Outline

Motivation

Proof of Concept

Approach

Evaluation

Conclusion

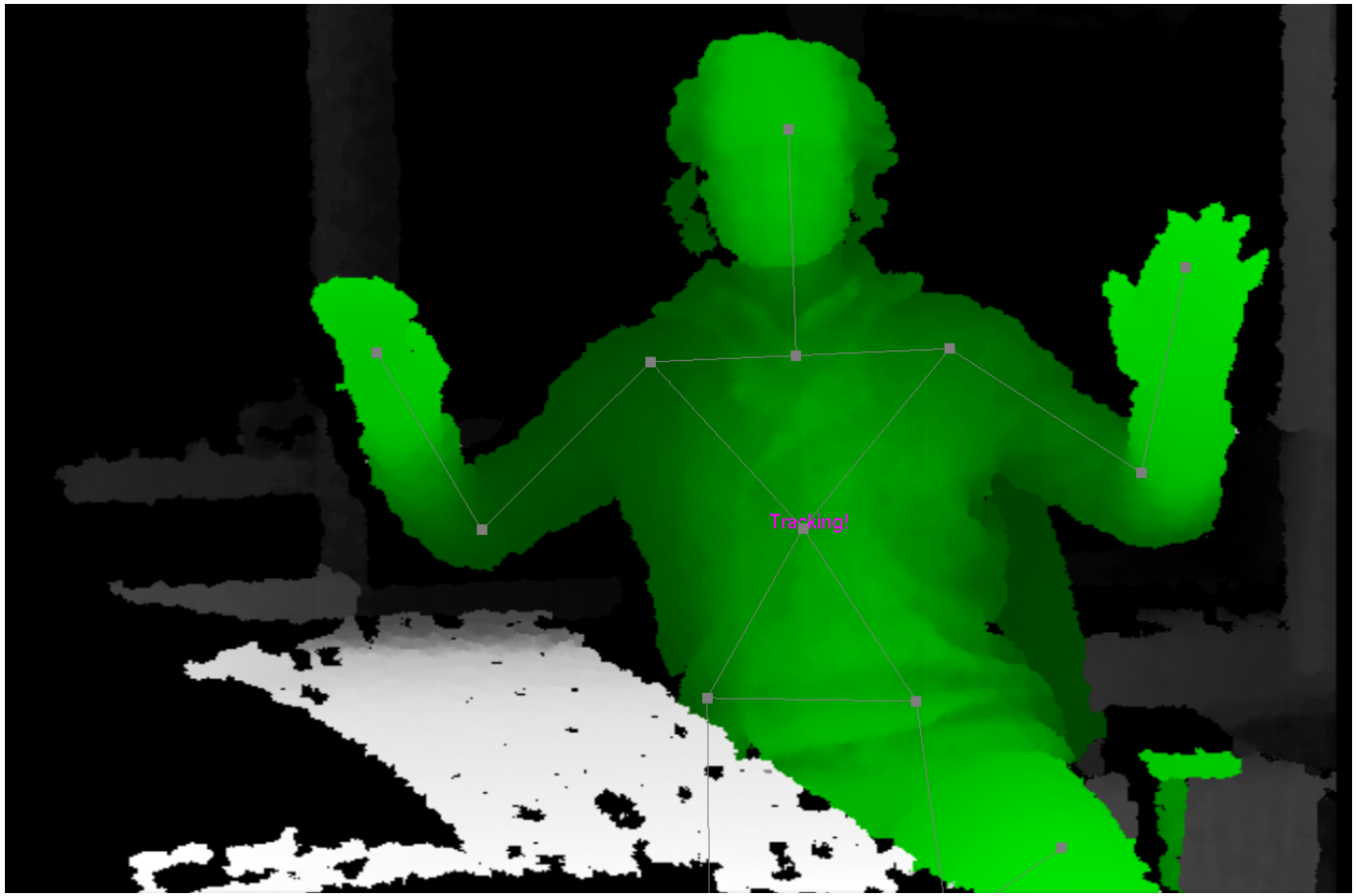
Proof of Concept

Overview

- Proof of concept implementation to show feasibility of the approach
- Natural Interaction via REST (NIREST)
  - Access on sensor details, recognized persons, and skeleton coordinates
  - Extracted on-the-fly from a video sensor
  - RESTful API with Linked Data resources
- Technology
  - Kinect sensor
  - OpenNI framework/NiTE middleware
  - Jena/Jersey libraries
  - Implemented as webapp for application server

Proof of Concept

Example



- Depth Image with Highlighted User and Skeleton

Proof of Concept

Example

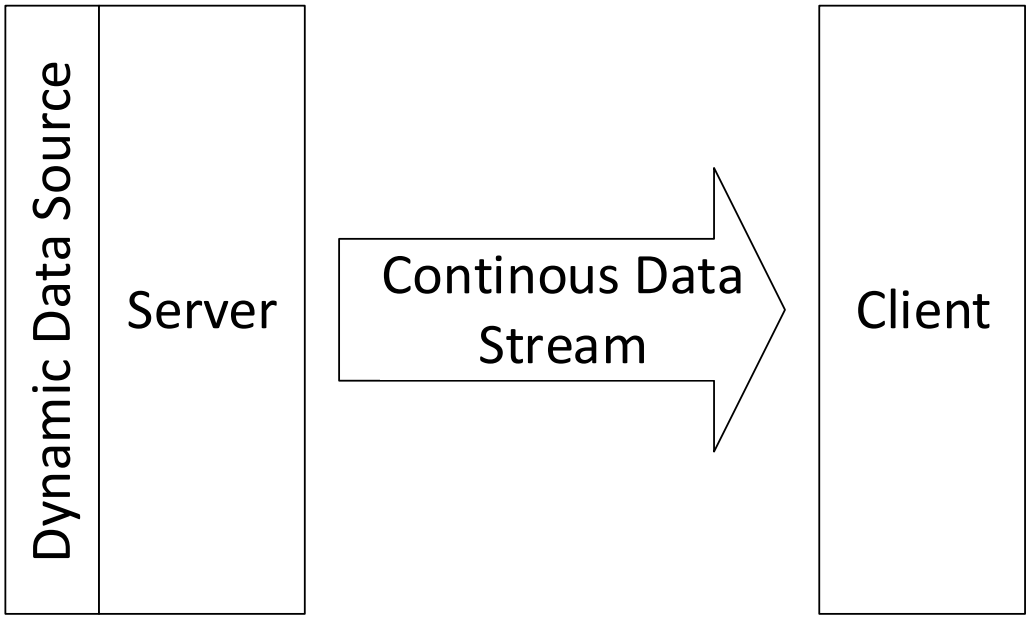
```
@prefix nirest: <http://vocab.arvida.de/2014/02/nirest/vocab#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .

<>
  a nirest:User ;
    nirest:centerOfMass [ a nirest:Coordinate3D ;
      nirest:x "-133.63106"^^xsd:float ;
      nirest:y "-27.113548"^^xsd:float ;
      nirest:z "2298.0164"^^xsd:float ] ;
    nirest:skeleton [ a nirest:Skeleton ;
      nirest:joint [ a nirest:RightHandJointPoint ;
        nirest:coordinate [ a nirest:Coordinate3D ;
          nirest:x "-2.150751"^^xsd:float ;
          nirest:y "-1.2886047"^^xsd:float ;
          nirest:z "2324.944"^^xsd:float ] ;
        nirest:orientationConfidence "1.0"^^xsd:float ;
        nirest:positionConvidence "1.0"^^xsd:float ] ;
...
```

- Example: User and Skeleton Tracking Data in Turtle Format
  - Resource e.g. http://[...]/device/0/user/1

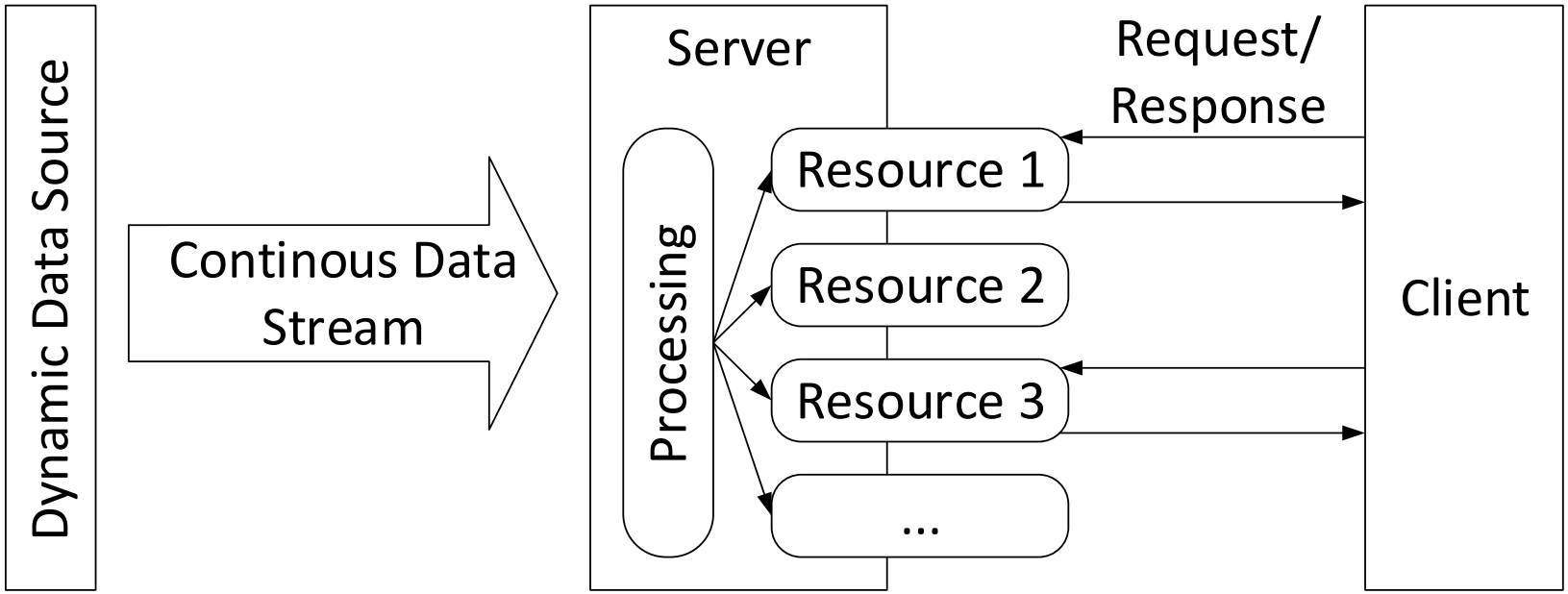
Outline

Approach
Architecture



- Dynamic data source exposed as a stream

Approach
Architecture



- Dynamic data source exposed with a REST API

Approach

Advantages

- Data Preparation
  - Data encapsulated as individual resources
  - (Pre-)processing shifted to the server
  - Client relieved from processing and identifying relevant data
- Example
  - Sensor recognizes objects in the video stream
  - Individual resource for each object

Approach

Advantages

- Data Selection
  - Reduction of communicated data
  - Clients limit requests to required information
  - Server discards irrelevant information
- Example
  - Sensor distinguishes people and objects
  - Only objects requested by the client

Approach

Advantages

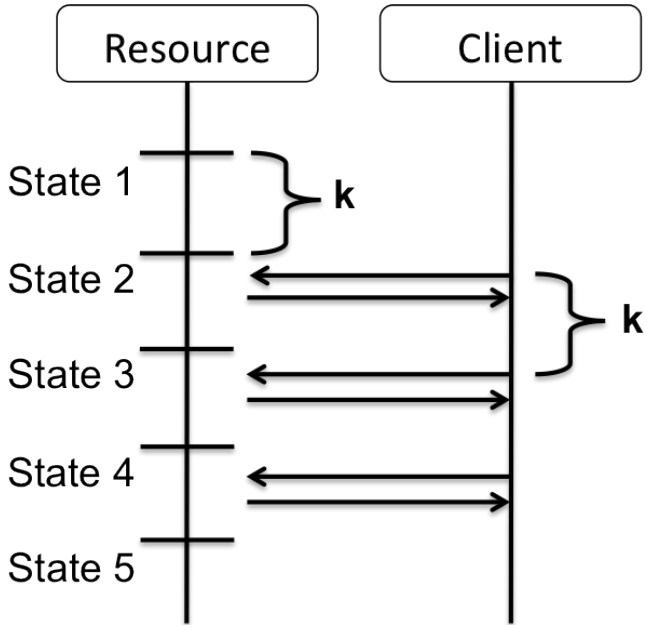
- Interoperability
  - Integration and eased use of different sources via Linked Data
  - Resources may provide links to other internal or external resources
- Example
  - Sensor supports face recognition
  - Data contains links to social network information

Approach

Advantages

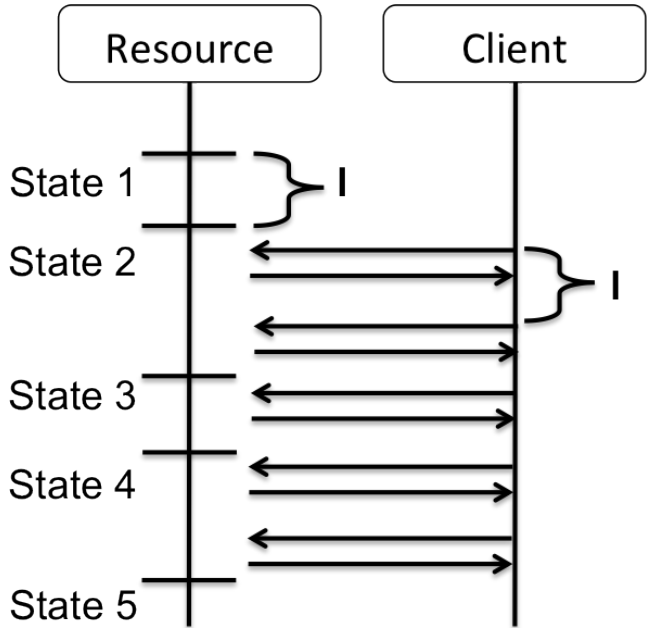
- Performance Decoupling
  - Client might not require updates in realtime
  - Client might not be able to process all updates in realtime
  - Individual resources and pull-based approach enable fine grained scaling
- Example
  - Sensor links to precise polygon data
  - Client retrieves the polygon data only once

Approach
Limitations



- Regular state update interval *k*
- Client request frequency must be at least *k* to retrieve all updates

Approach
Limitations



- Inconsistent state update interval, minimal interval *i*
- Client request frequency must be at least *i* to retrieve all updates
- Client may receive duplicates of a state

Outline

Evaluation

Terms

- Frequency
  - Measured in events per second
  - Unit: Hertz
  - 1 Event / Second = 1 Hertz (Hz)
- Jitter
  - Unexpected deviation from a periodic signal
  - 10 Hz (expected) - 7 Hz (measured) = 3 Hz Jitter

Evaluation

Setup

- Preliminary evaluation on performance of the implementation
- Server setup
  - Microsoft Kinect v1
  - Apache Tomcat with deployed NIREST
  - Switched gigabit network
- Client setup
  - Second independent computer
  - Switched gigabit network of the server
  - Requests representations of a person in front of the sensor
  - Complete HTTP request/response per request

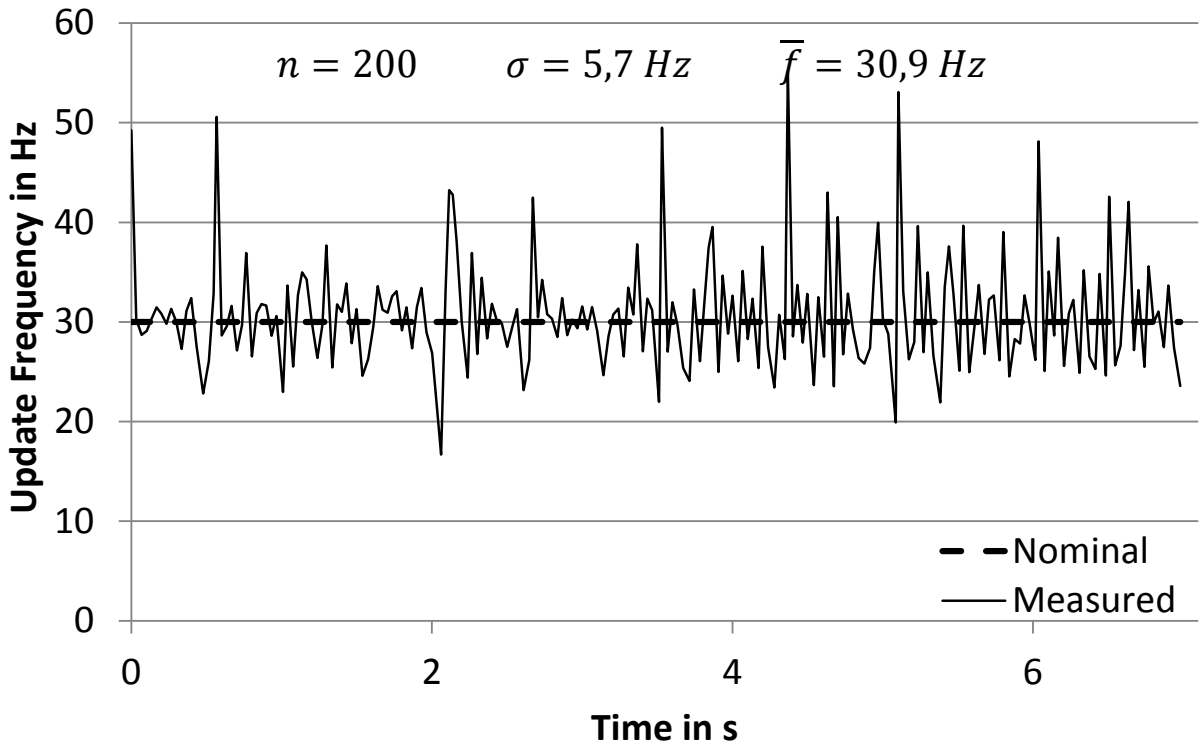
Evaluation

Measurement 1

- Server side update frequencies
- Settings
  - Microsoft Kinect v1 with 30 Hz nominal update frequency
  - Sample of 200 updates
  - Updates measured server side
- Measurement 1
  - 30.9 Hz average update frequency
  - 5.9 Hz standard deviation

Evaluation

Measurement 1



- Server side update frequencies

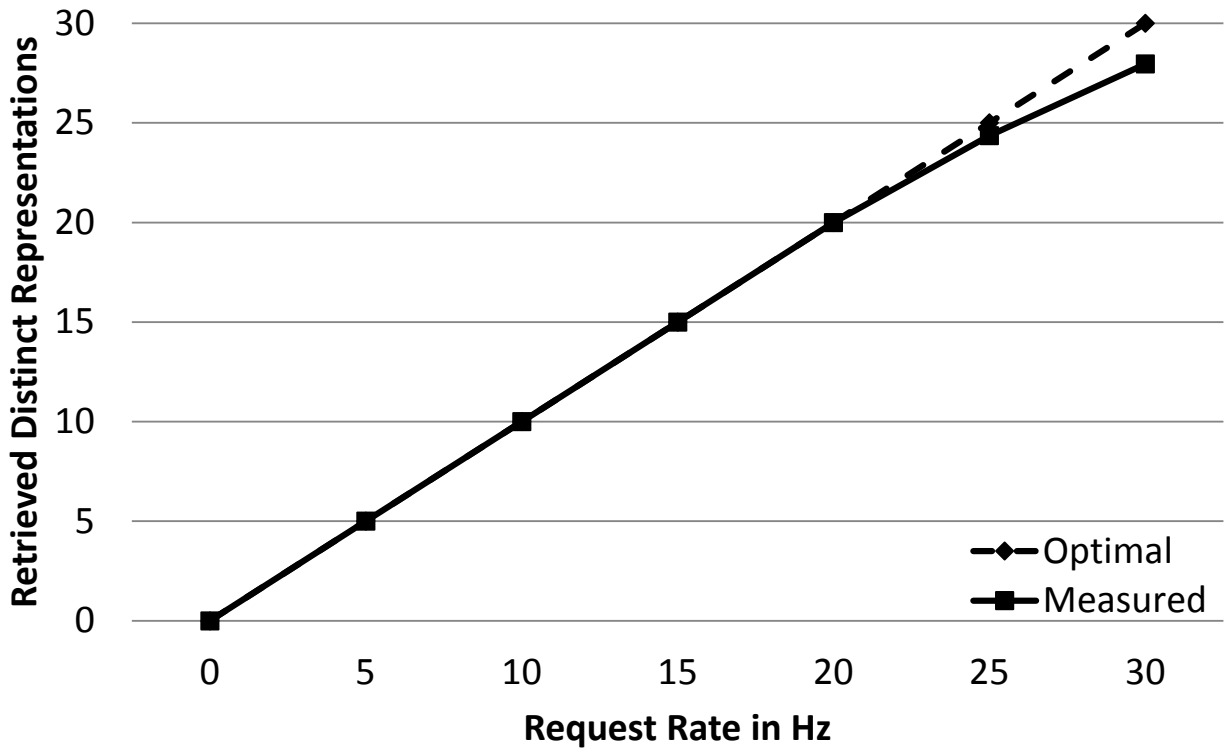
Evaluation

Measurement 2

- Distinct representations for different client request frequencies
- Settings
  - Microsoft Kinect v1 with 30 Hz nominal update frequency
  - Different request frequencies between 5 Hz and 30 Hz
  - Theoretical optimal retrieval with 30 Hz request frequency
- Measurement 2
  - 5 - 25 Hz request frequency
    - Optimal amount of distinct representations
  - 25 - 30 Hz request frequency
    - Some representations missing
    - Average of 28 distinct representations at 30 Hz

Evaluation

Measurement 2



- Distinct representations for different client request frequencies

Evaluation

Discussion

- Highly dynamic RESTful communication pattern is achievable
- Optimization potential
  - Reduction of the jitter in the server side update frequencies
  - No big optimization by increasing the request rate
- More homogeneous update rates
  - Retrieval of all information
  - No increase the retrieval overhead required

Outline

Conclusion

- Outlined advantages of a combination of
  - RESTful API
  - Linked Data
  - Highly dynamic data source
- Proof of concept implementation
- Highly dynamic RESTful communication pattern
  - Feasibility indicated by preliminary results
  - Optimization potential identified

?