<QuakeML>

QuakeML

XML concepts for a European seismological data exchange infrastructure

Fabian Euchner (ETH Zürich)
Danijel Schorlemmer (USC, Los Angeles)
and the *QuakeML working group*
(ETH, GFZ, USC, USGS, IRIS)

QuakeML – Design Principles

- Intended to cover a broad range of seismological fields
- Support real-time data transfer of seismic events
- Designed as data exchange format, independent of further persistent storage
- Modular approach:
  - Basic event data
  - Moment tensor
  - Metadata infrastructure
  - ...
- Written from scratch, no adaption of existing (DB-) schema, uses full XML flexibility
- Community-driven development of standards
  - Documents are first discussed in working group, then subjected to **Request for Comments** process
  - Sequence of document maturity levels: Working Draft, Proposed Recommendation, Recommendation

QuakeML – Definition and Standardization

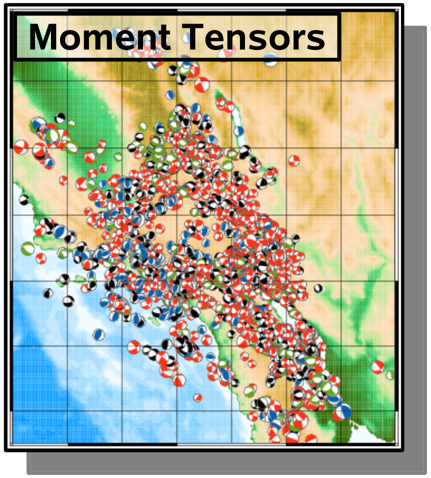
- SED, GFZ and others are in the process of defining a set of QuakeML 1 elements (6 meetings held so far)
- Request for Comments document available April 2007
- Further QuakeML development will be based at SED
- SeisComP 3 and ZMAP++ use QuakeML for internal communication
- Already developed toolchain for automated data structure generation and code generation
- Standard UML 2.0 schema description

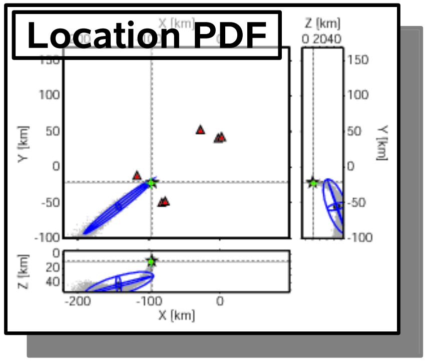
QuakeML – Development Roadmap

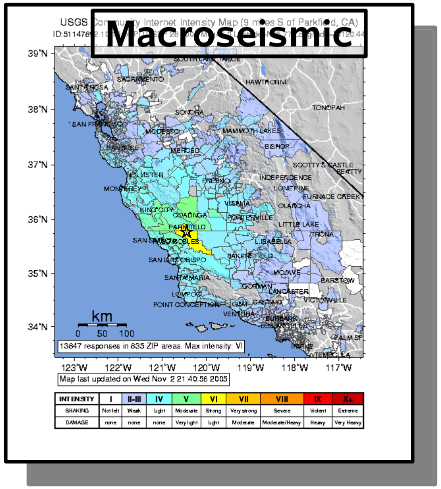
QuakeML development roadmap

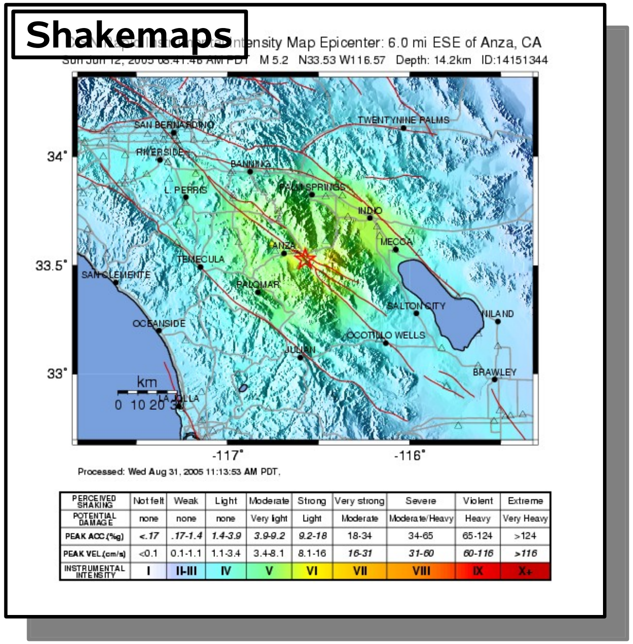
Basic event description (incl. moment tensors)	RFC starts April 2007
Metadata infrastructure	RFC starts Summer 2007
Inventory	RFC starts ?

Further development: location PDF (Nlloc), macroseismic information, slip distribution, shakemaps, ...


Moment Tensors


Location PDF

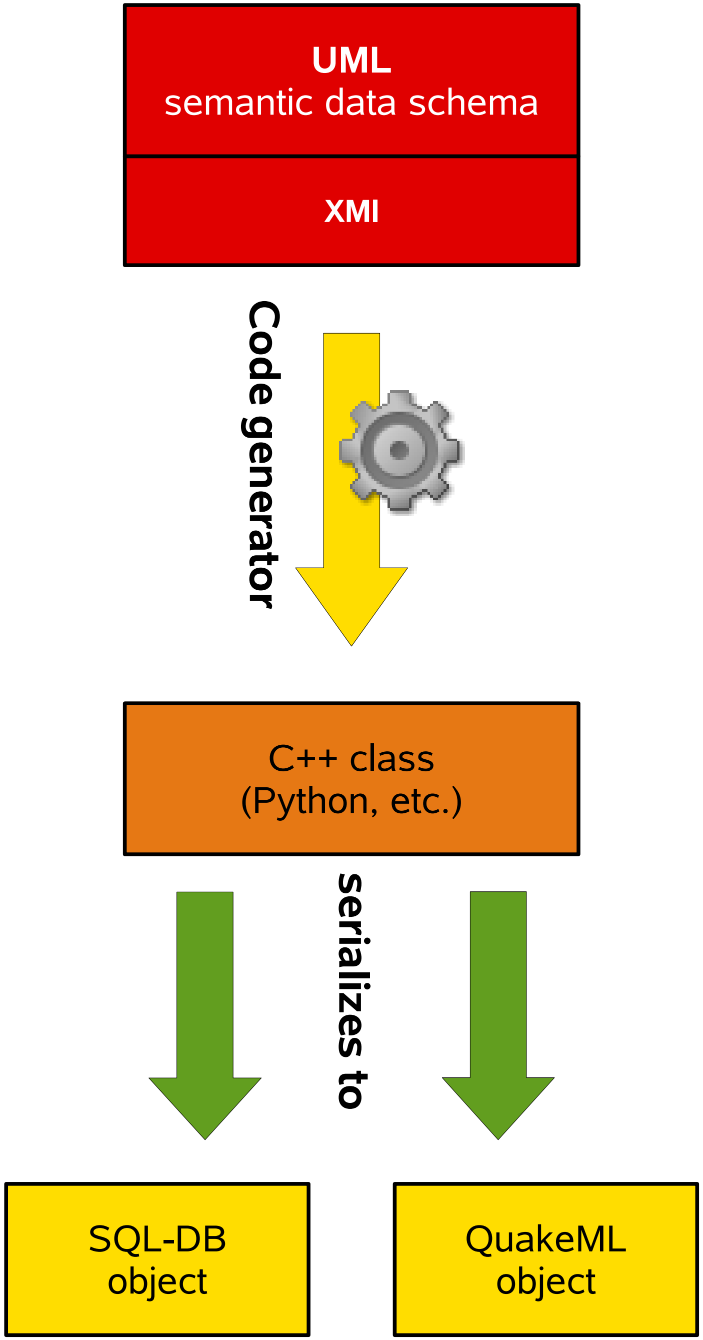

Macroseismic
USGS Community Internet Intensity Map (9 miles S of Parkfield, CA)
39°N
38°N
37°N
36°N
35°N
34°N
123°W
122°W
121°W
120°W
119°W
118°W
117°W
km
0 50 100
13647 responses in 635 ZIP areas. Max intensity: VI
Map last updated on Wed Nov 2 21:40:56 2005
INTENSITY
SHAKING
DAMAGE


Shakemaps

<QuakeML>

QuakeML – Tools and Services

- QuakeML data model available as UML class diagram (and XMI)
- We will maintain and distribute a software library for QuakeML:
    - Use XMI for automated code generation
    - Library native in C++, can be used from other object-oriented languages (Python, Java) using wrappers
    - Objects can be serialized as XML (QuakeML) or SQL



UML
semantic data schema

XMI

Code generator

C++ class
(Python, etc.)

serializes to

SQL-DB
object

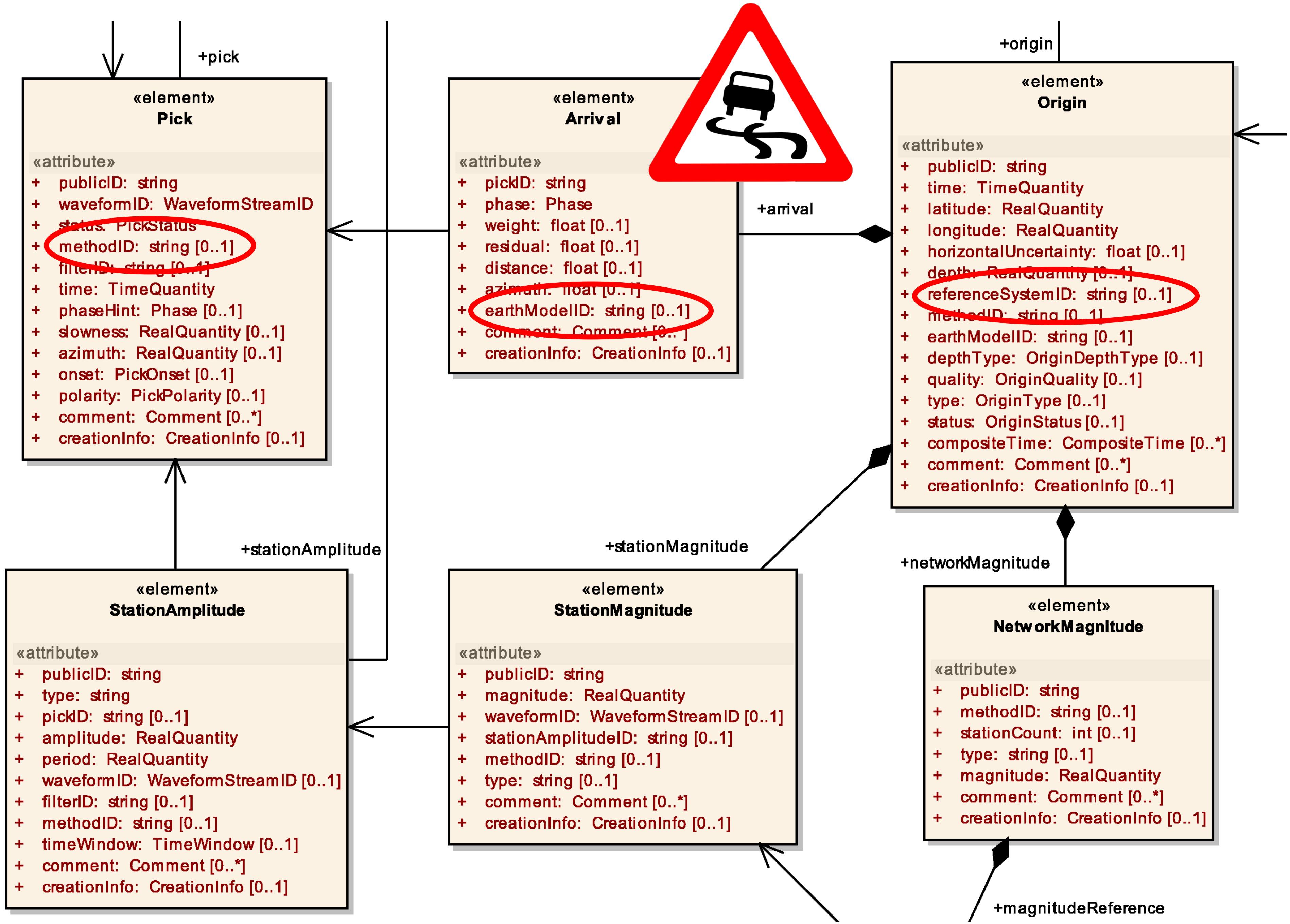
QuakeML
object

QuakeML – Code Generator

- creates C++ class library, XML Schema, and SQL DB schema from UML/XMI data model
- creates Python wrappers using SWIG toolkit (wrappers for other OO languages can be added)
- written in Python, uses Cheetah template engine
- open source (GPL)
- originally developped by GFZ as a part of SeisComP, many dependencies on SeisComP components
- a stand-alone version will be provided with the QuakeML standard documents and Schema
- QuakeML code generator/class library enable users to work with QuakeML without having to write own code
- code generator toolchain allows to create own, site-specific extensions of QuakeML

<QuakeML>

Metadata/Data Problem



+pick

«element»
Pick

«attribute»
- + publicID: string
- + waveformID: WaveformStreamID
- + status: PickStatus
- + methodID: string [0..1]
- + filterID: string [0..1]
- + time: TimeQuantity
- + phaseHint: Phase [0..1]
- + slowness: RealQuantity [0..1]
- + azimuth: RealQuantity [0..1]
- + onset: PickOnset [0..1]
- + polarity: PickPolarity [0..1]
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

«element»
Arrival

«attribute»
- + pickID: string
- + phase: Phase
- + weight: float [0..1]
- + residual: float [0..1]
- + distance: float [0..1]
- + azimuth: float [0..1]
- + earthModelID: string [0..1]
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

+arrival

+origin

«element»
Origin

«attribute»
- + publicID: string
- + time: TimeQuantity
- + latitude: RealQuantity
- + longitude: RealQuantity
- + horizontalUncertainty: float [0..1]
- + depth: RealQuantity [0..1]
- + referenceSystemID: string [0..1]
- + methodID: string [0..1]
- + earthModelID: string [0..1]
- + depthType: OriginDepthType [0..1]
- + quality: OriginQuality [0..1]
- + type: OriginType [0..1]
- + status: OriginStatus [0..1]
- + compositeTime: CompositeTime [0..*]
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

+stationAmplitude

+stationMagnitude

+networkMagnitude

«element»
StationAmplitude

«attribute»
- + publicID: string
- + type: string
- + pickID: string [0..1]
- + amplitude: RealQuantity
- + period: RealQuantity
- + waveformID: WaveformStreamID [0..1]
- + filterID: string [0..1]
- + methodID: string [0..1]
- + timeWindow: TimeWindow [0..1]
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

«element»
StationMagnitude

«attribute»
- + publicID: string
- + magnitude: RealQuantity
- + waveformID: WaveformStreamID [0..1]
- + stationAmplitudeID: string [0..1]
- + methodID: string [0..1]
- + type: string [0..1]
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

«element»
NetworkMagnitude

«attribute»
- + publicID: string
- + methodID: string [0..1]
- + stationCount: int [0..1]
- + type: string [0..1]
- + magnitude: RealQuantity
- + comment: Comment [0..*]
- + creationInfo: CreationInfo [0..1]

+magnitudeReference

<QuakeML>

Data Exchange Infrastructure & Resource Identifiers

Requirements for seismological information exchange in a global network

- unambiguous, persistent identifiers of networked resources
- rich standardized metadata description (Dublin Core)
- resource discovery by standardized query gateways
- information retrieval using standardized services and protocols

QuakeML supports unique, location-independent identifiers for resources in a seismological network (in URI format)

```
smi://<authority-id>/<resource-key>[#<local-id>]
```

URI schema `smi://` stands for seismological meta-information

Example: SED autopicker

```
smi://ch.ethz.sed/Software/Picker/Autopicker/6.0.5
```

Data Exchange Infrastructure & Resource Identifiers

A **registry** is a resource that stores and provides metadata about other resources

- provides a Web Service which resolves identifiers into a RDF metadata description
- vocabulary will be based on Dublin Core, with extensions for seismological resources

Metadata will cover the resource's

- identity (name and description)
- curation (who maintains it?)
- content (what kind of information?)
- data quality
- service (how to retrieve the resource?)

Registry
Webservice

QuakeML Resource Metadata

The RDF metadata contain information on how to retrieve the resource data, e.g., URLs, pointers to Web Service descriptions (WSDL), etc.

- if the resource data are in QuakeML format, they can be retrieved via registries with a short-cut identifier

  `quakeml://<authority-id>/<resource-key>[#<local-id>[?<parameters>]]`

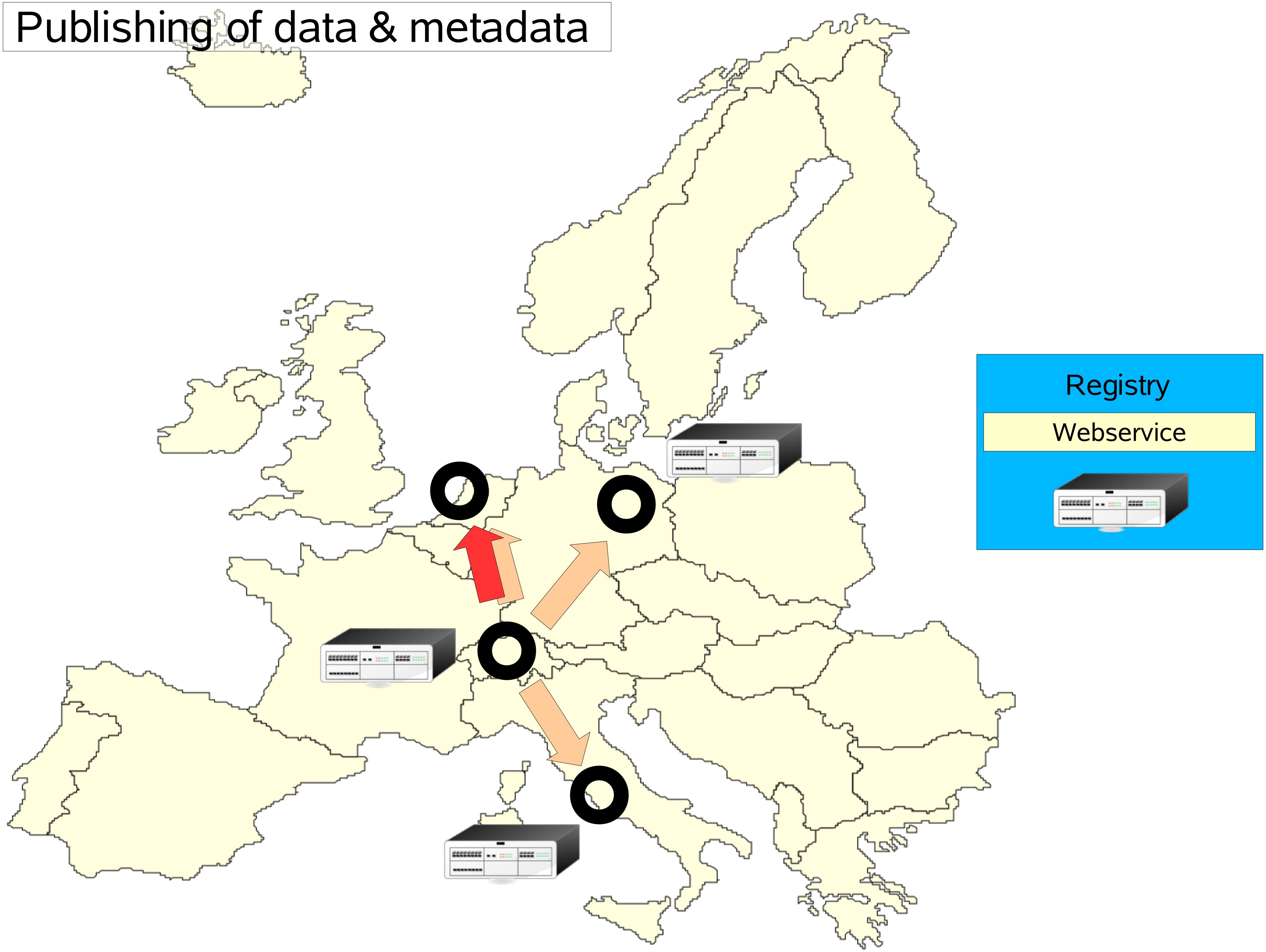
- URI schema `quakeml://` indicates that resource data are available in QuakeML format

- additional <parameters> may be required to retrieve the resource data (e.g., waveform data for a time interval)

- <parameters> part is in URL parameter format: `par1=value1&par2=value2&...`

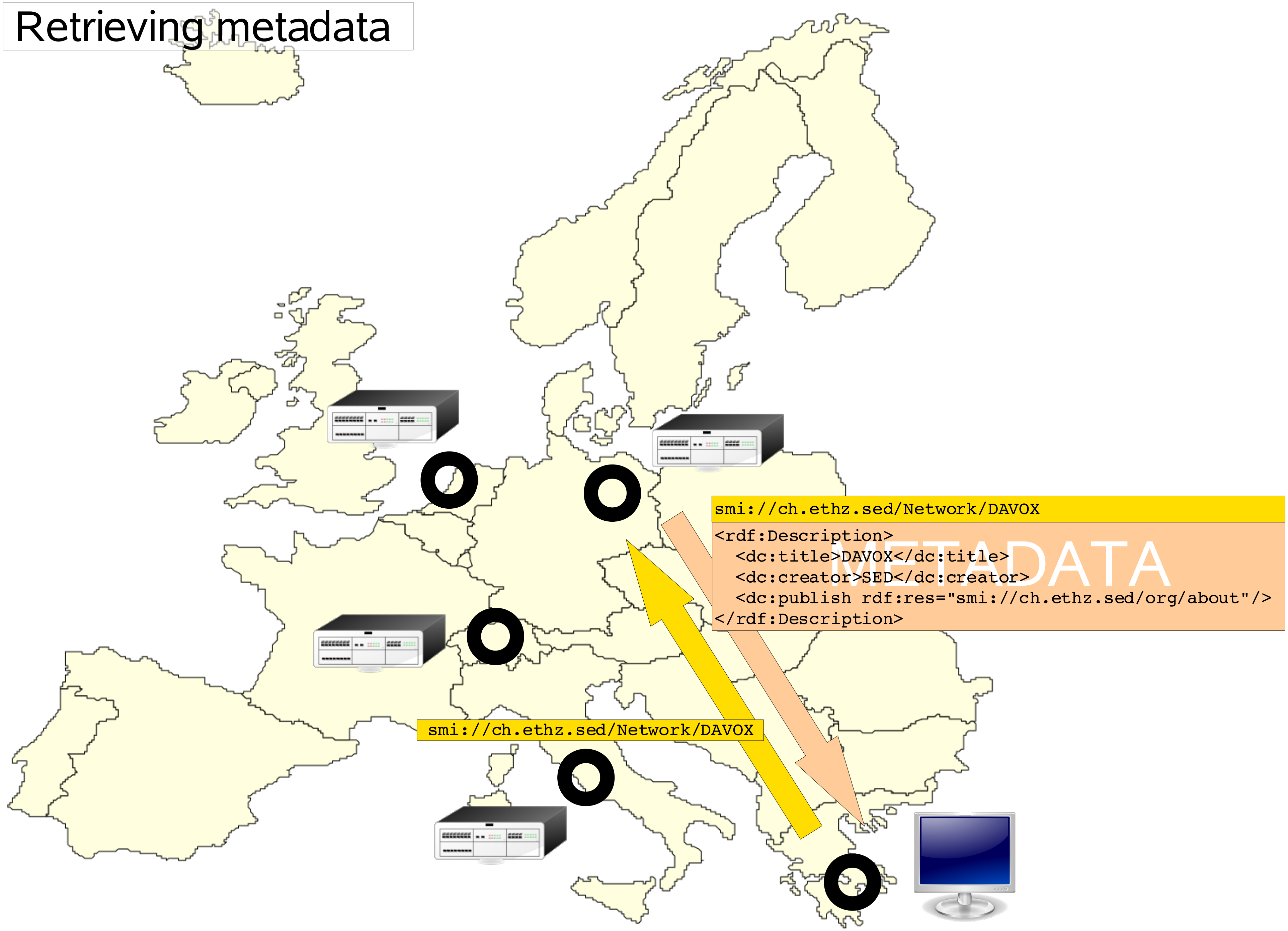
Metadata & Data Exchange

Publishing of data & metadata



Metadata & Data Exchange

Retrieving metadata



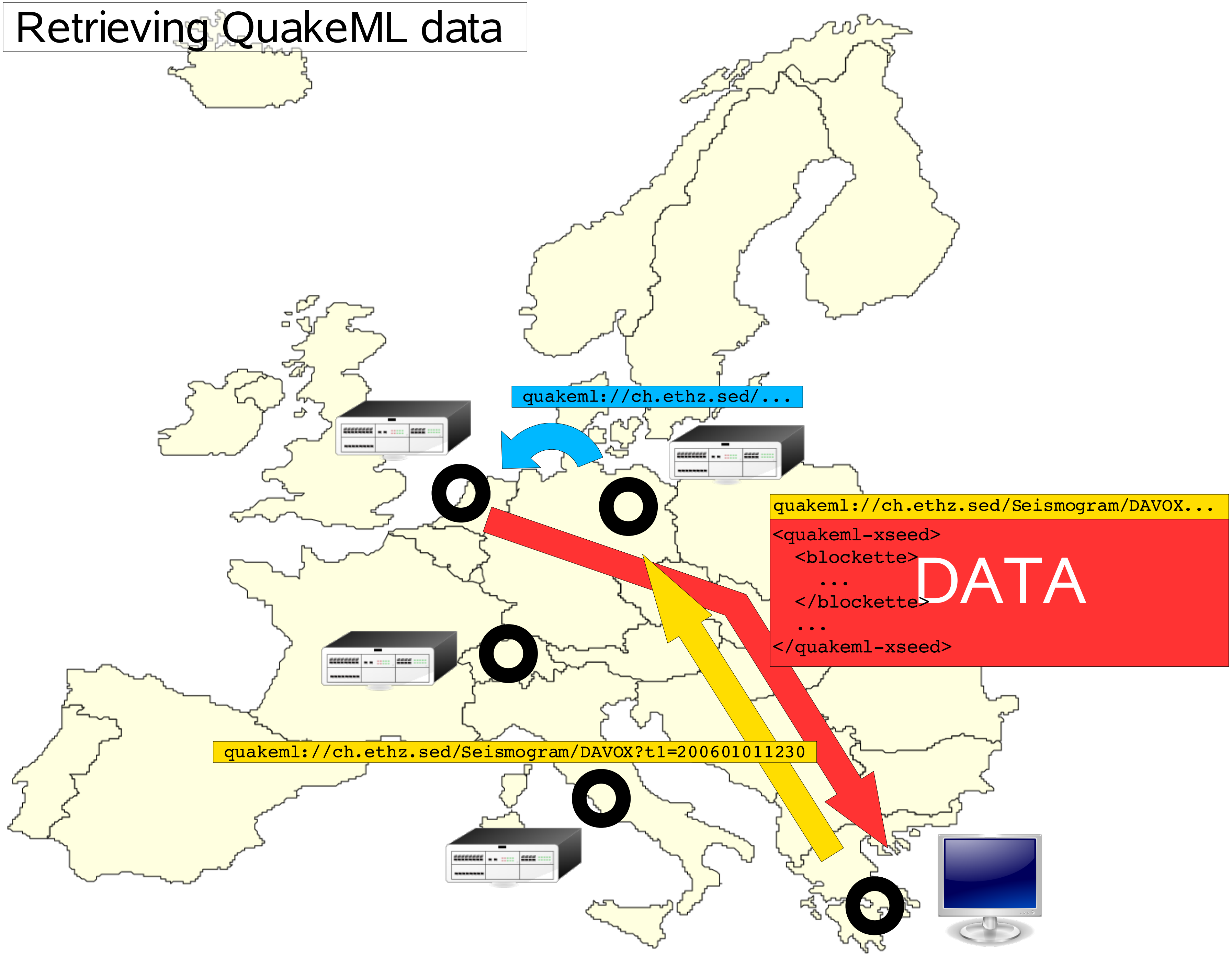
smi://ch.ethz.sed/Network/DAVOX

smi://ch.ethz.sed/Network/DAVOX

```
<rdf:Description>
  <dc:title>DAVOX</dc:title>
  <dc:creator>SED</dc:creator>
  <dc:publish rdf:res="smi://ch.ethz.sed/org/about"/>
</rdf:Description>
```

Metadata & Data Exchange

Retrieving QuakeML data



quakeml://ch.ethz.sed/...

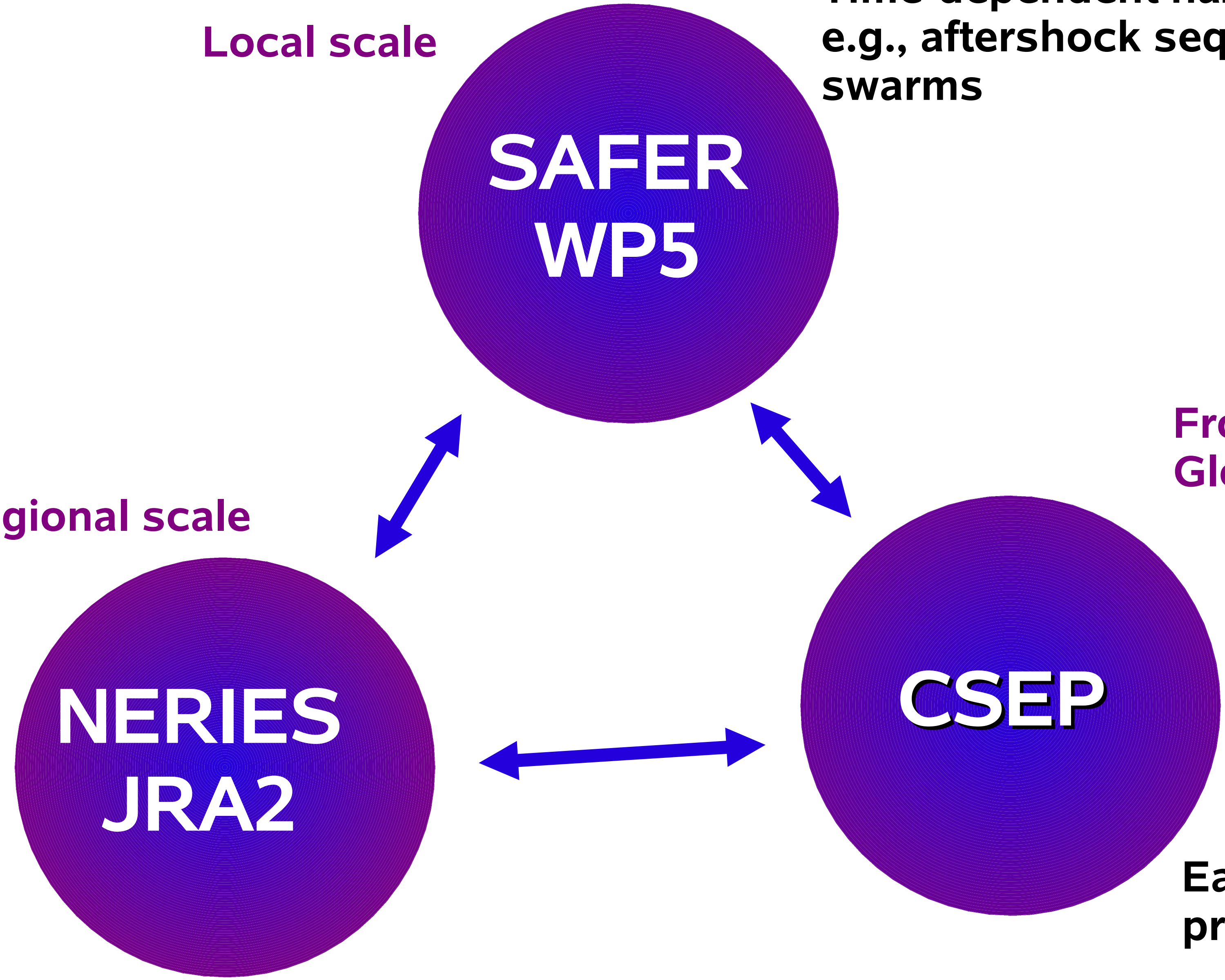
quakeml://ch.ethz.sed/Seismogram/DAVOX...

```
<quakeml-xseed>
  <blockette>
    ...
  </blockette>
  ...
</quakeml-xseed>
```

DATA

quakeml://ch.ethz.sed/Seismogram/DAVOX?t1=200601011230

QuakeML – In International Projects



Local scale

Time-dependent hazard for, e.g., aftershock sequences, swarms

Regional scale

Time-dependent hazard on nationwide/european scale

From Local to Global scale

Earthquake predictability

<QuakeML>

QuakeML Development Team

QuakeML core development team (quakeml@sed.ethz.ch)

ETH Zürich

- Fabian Euchner
- Philipp Kästli
- Silvio Maraini
- Stefan Wiemer
- Jochen Wössner

GFZ Potsdam

- Jan Becker
- Winfried Hanka
- Andres Heinloo
- Joachim Saul
- Bernd Weber

USC, Los Angeles

- Danijel Schorlemmer

USGS/NEIC, Denver

- Ray Buland

IRIS DMC, Seattle

- Linus Kamb

Contact Us

- Check our website

  http://www.quakeml.ethz.ch
  http://www.quakeml.org (upcoming)

- Write us

  quakeml@sed.ethz.ch

  - for participating in the request for comments process: Send us your email-address and you will receive the RFC documents
  - for questions
  - for suggestions