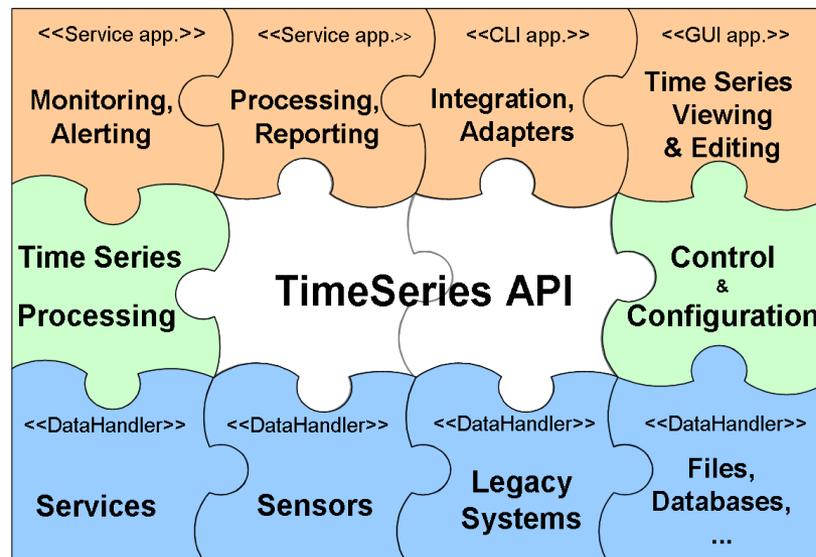# The New Time Series Toolbox

## Next Generation of Sensor Web and Time Series Processing

Bojan Božić, MSc

# The Time Series Toolbox

- A set of applications and software components with a common API
- High-level Framework allowing efficient access to, processing, and presentation of semantically enriched time series
- For building services and applications
- Works with time series data (record, store, process, publish ...)

| <<Service app.>> | <<Service app.>> | <<CLI app.>> | <<GUI app.>> |
|---|---|---|---|
| Monitoring, Alerting | Processing, Reporting | Integration, Adapters | Time Series Viewing & Editing |

Time Series Processing — TimeSeries API — Control & Configuration

| <<DataHandler>> | <<DataHandler>> | <<DataHandler>> | <<DataHandler>> |
|---|---|---|---|
| Services | Sensors | Legacy Systems | Files, Databases, ... |

# Why do you need a Time Series Toolbox?

- Components are easy to extend and to replace by alternatives (e.g. in order to improve performance, to access new types of data, or to assure conformance with emerging standards)

- Components can be chained to provide more sophisticated processing capabilities or used in parallel to provide alternative means of accessing, storing, and presenting data

- The control components allow dynamic re-configuration of components and data flow within applications

- Provided applications can be either used as educational material by developers, or used as they are by end-users, or extended and customized to meet particular requirements

# Components

- Sensor configuration and access to sensor data

- Read and write access to time series stored in services, files, and databases

- Service interfaces

- Processing and annotation of time series

- Dynamic control of data flow, and the behavior of data handlers and processing components
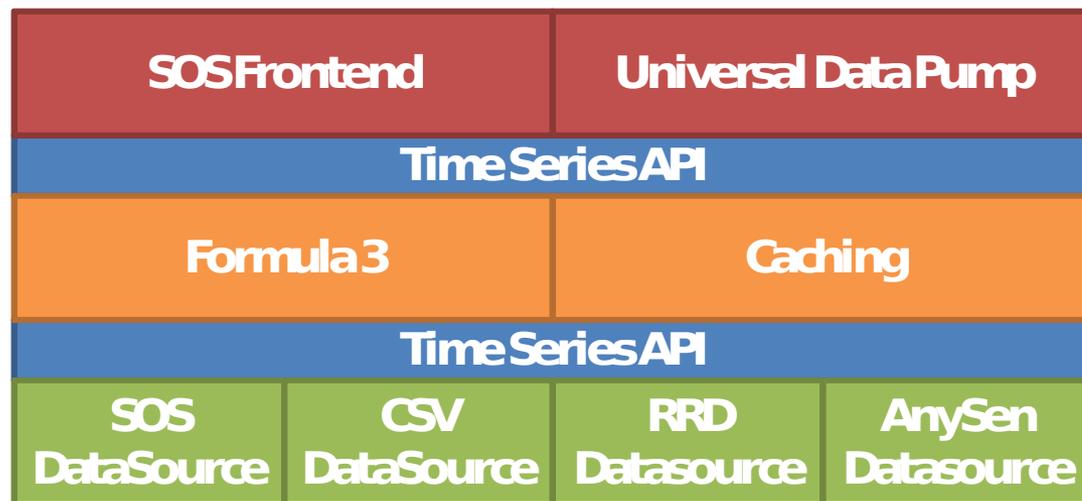
# Design Goals

- Modular Architecture: use the components you like or need, not everything or nothing

- Standards-based: e.g. Sensor Observation Service

- Flexible Design: from simple float values to complex data structures

- Extensible: well-defined interfaces allow easy extension

# Environment / Requirements

- Main language: Java 1.6

- One component written in Python ($\rightarrow$ Jython)

- Can be used in all types of applications:
    - Web services (J2EE, etc.)
    - Command-line applications
    - Desktop GUI applications
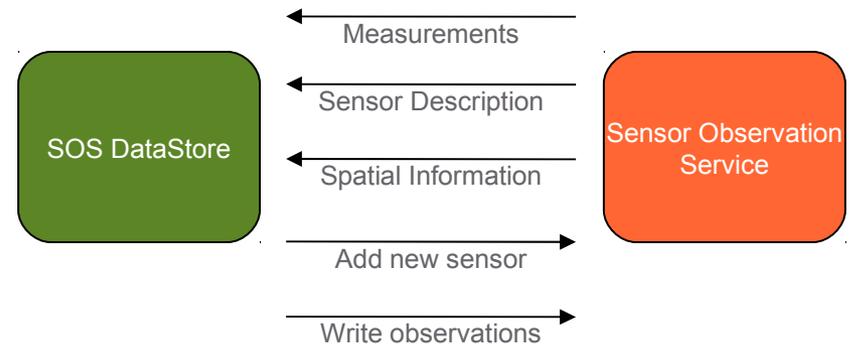
# Architecture

- **Data Handlers**

- Processing and Control Components

- Applications and Services

| SOS Frontend | Universal Data Pump |
|---|---|
| **Time Series API** | |
| Formula 3 | Caching |
| **Time Series API** | |

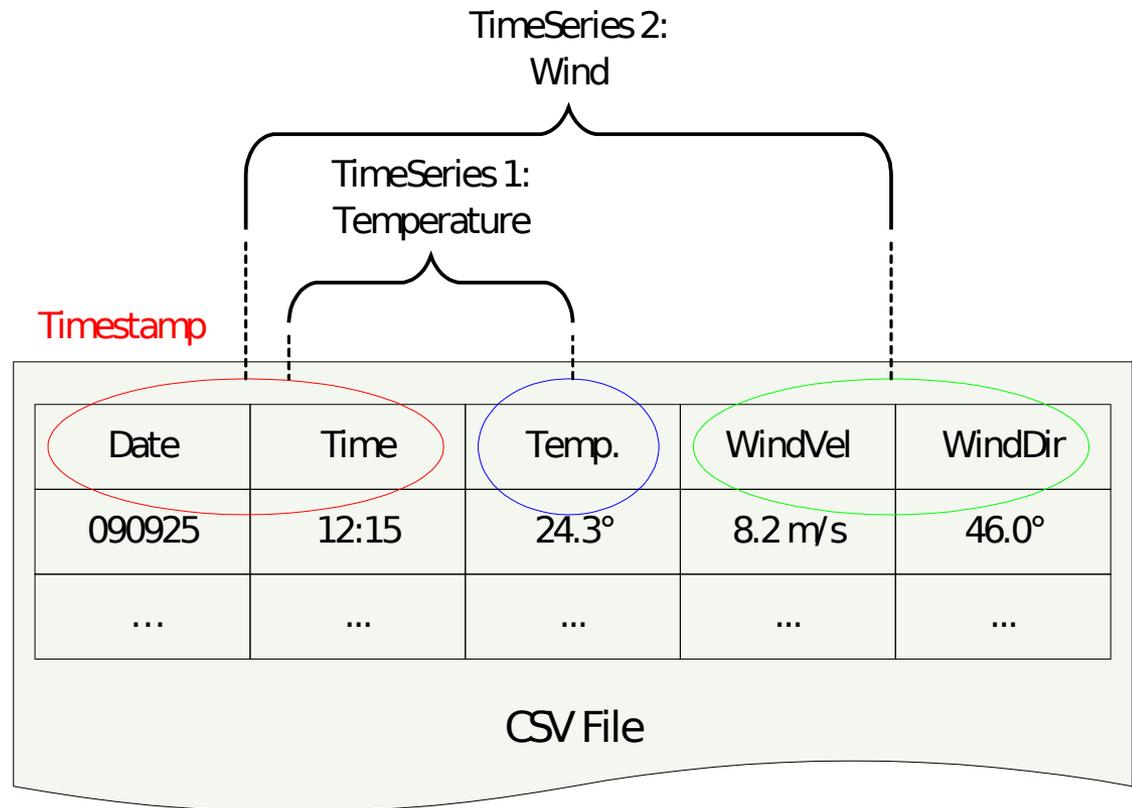| SOS DataSource | CSV DataSource | RRD Datasource | AnySen Datasource |
|---|---|---|---|

# SOS DataHandler

- Accessing Sensor Observation Services

- Read: Observations, SensorML, Features

- Write: Observations, RegisterSensor

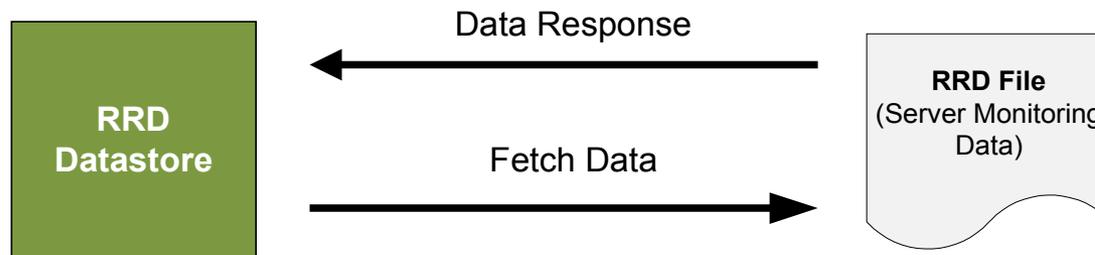- Flexible Parser (e.g. SOS 1.0 / 0.31)

```
                    ◄──── Measurements ────
┌──────────────┐    ◄──── Sensor Description ────    ┌──────────────┐
│              │                                      │              │
│ SOS DataStore│    ◄──── Spatial Information ────    │Sensor Observation│
│              │    ──── Add new sensor ────►         │   Service    │
└──────────────┘    ──── Write observations ────►     └──────────────┘
```

# CSV DataHandler

- Access Comma Separated Value Files

- Read and write access

- Flexible configuration
    - Separators
    - Formats
    - Values

TimeSeries 2:
Wind

TimeSeries 1:
Temperature

Timestamp

| Date | Time | Temp. | WindVel | WindDir |
|------|------|-------|---------|---------|
| 090925 | 12:15 | 24.3° | 8.2 m/s | 46.0° |
| … | ... | ... | ... | ... |

CSV File

# RRD DataHandler

- Round Robin Database (http://oss.oetiker.ch/rrdtool/)

- Currently read-only access to data

- Integration of existing RRD files (e.g. from system monitoring applications)

```
          Data Response
    ┌──────────────◄──────────────┐
    │    RRD                       │   RRD File
    │  Datastore                   │ (Server Monitoring
    │                              │      Data)
    └──────────────►──────────────┘
            Fetch Data
```
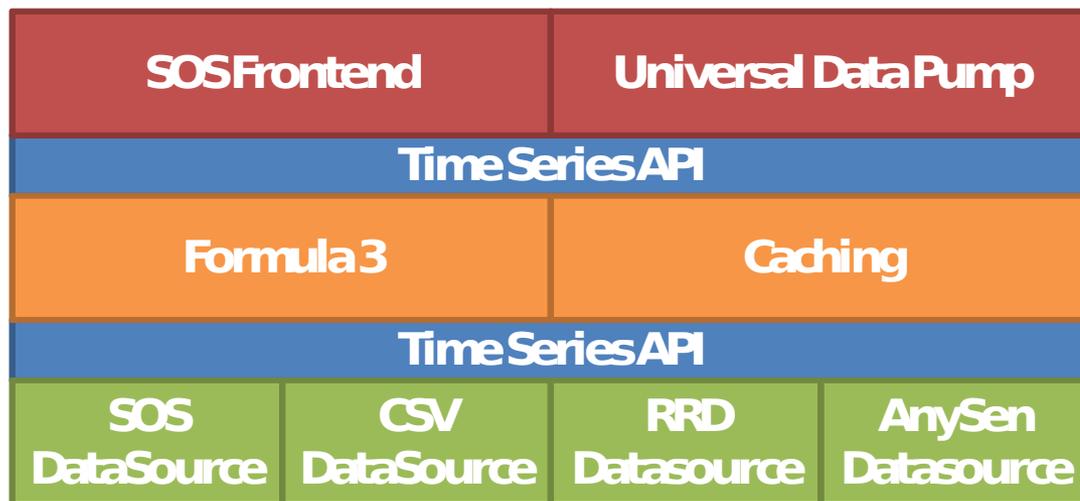
# AnySen DataHandler

- Data acquisition directly from a sensor

- Designed for serial streaming sensors

- Flexible configuration of parser

- Plug & measure – get config from repository

# Architecture

- Data Handlers

- **Processing and Control Components**

- Applications and Services

| SOS Frontend | | Universal Data Pump | |
|---|---|---|---|
| Time Series API | | | |
| Formula 3 | | Caching | |
| Time Series API | | | |
| SOS DataSource | CSV DataSource | RRD Datasource | AnySen Datasource |

# Formula 3

- "Time Series Processor"

- Read time series data from DataSources via TSAPI

- Process data:
  - `< [n] * 2 >`
  - `< [n-2 .. n].sum >`
  - `< (t-3 hours .. t].mean > every 30 mins`

- Output time series data via TSAPI
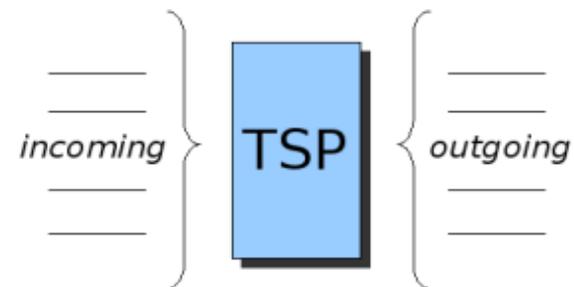
# Formula 3

- Functionality:
    - Calculation (arithmetic)
    - Parameters for formulas
    - Time patterns, slot/range selection
    - Conditions
    - Predefined functions, user defined functions

- Example use cases:
    - Aggregation, mean values, etc.
    - Filtering, classification

# Formula 3 - Implementation

- Design goal: usable as a „standalone" library on major platforms (Java, .NET, "native")

- Implemented in Python

- TS Toolbox contains Formula 3 running on Jython

# Caching

- Store time series data in local database

- Temporary storage

- Use cases: caching, prefetching, local storage, etc.
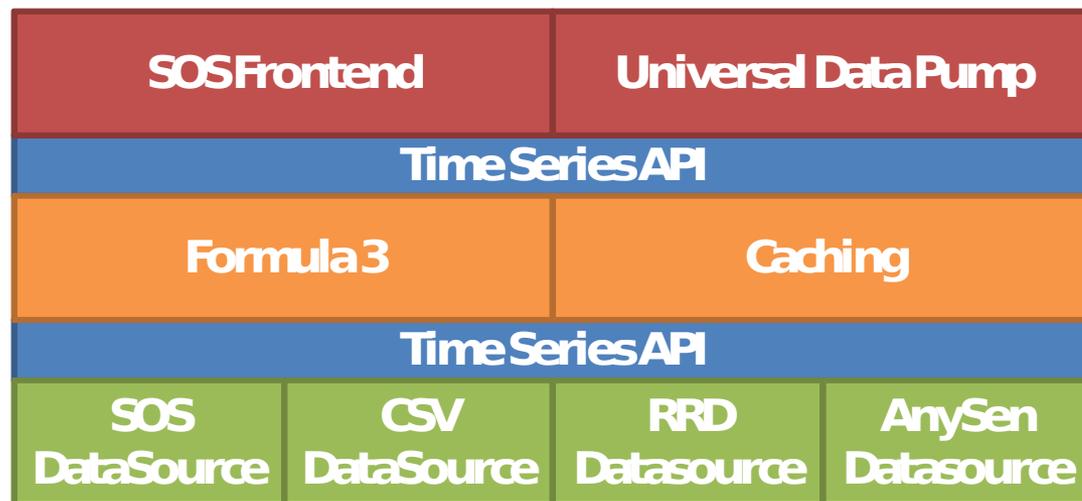
- Currently implements configurable prefetching

```
┌─────────────┐                                          ┌───────────┐
│             │      Retrieve serialized TimeSeries      │           │
│   Caching   │ ◄────────────────────────────────────   │  Data base│
│  DataStore  │                                          │(Hibernate │
│             │      Serialize and write TimeSeries      │compatible)│
│             │ ────────────────────────────────────►   │           │
└─────────────┘                                          └───────────┘
```

# Pipe Oriented Architecture (POA)

- Objectives:
  - Main operation is transport and processing of time series
  - Pipes for connecting data sources and data sinks
  - Pipes can also incorporate processing components (like F3)
  - Managers for control and coordination of processing components, data handlers, and pipes

- Managers:
  - Device Manager: manages data handlers
  - System Manager: Analyses the current system configuration
  - Pipe Manager: Builds pipes and passes events

# Architecture

- Data Handlers

- Processing and Control Components

- **Applications and Services**

| SOS Frontend | | Universal Data Pump | |
|---|---|---|---|
| Time Series API | | | |
| Formula 3 | | Caching | |
| Time Series API | | | |
| SOS DataSource | CSV DataSource | RRD Datasource | AnySen Datasource |

# SOS Frontend

- Sensor Observation Service Frontend to TSAPI

- Serve data from multiple data sources

- On-the-fly processing using Formula 3

- Local caching/prefetching of data

- Based on 52° North SOS

- SensorML generation/transformation with XSLT

# SOS Frontend: Application Schema
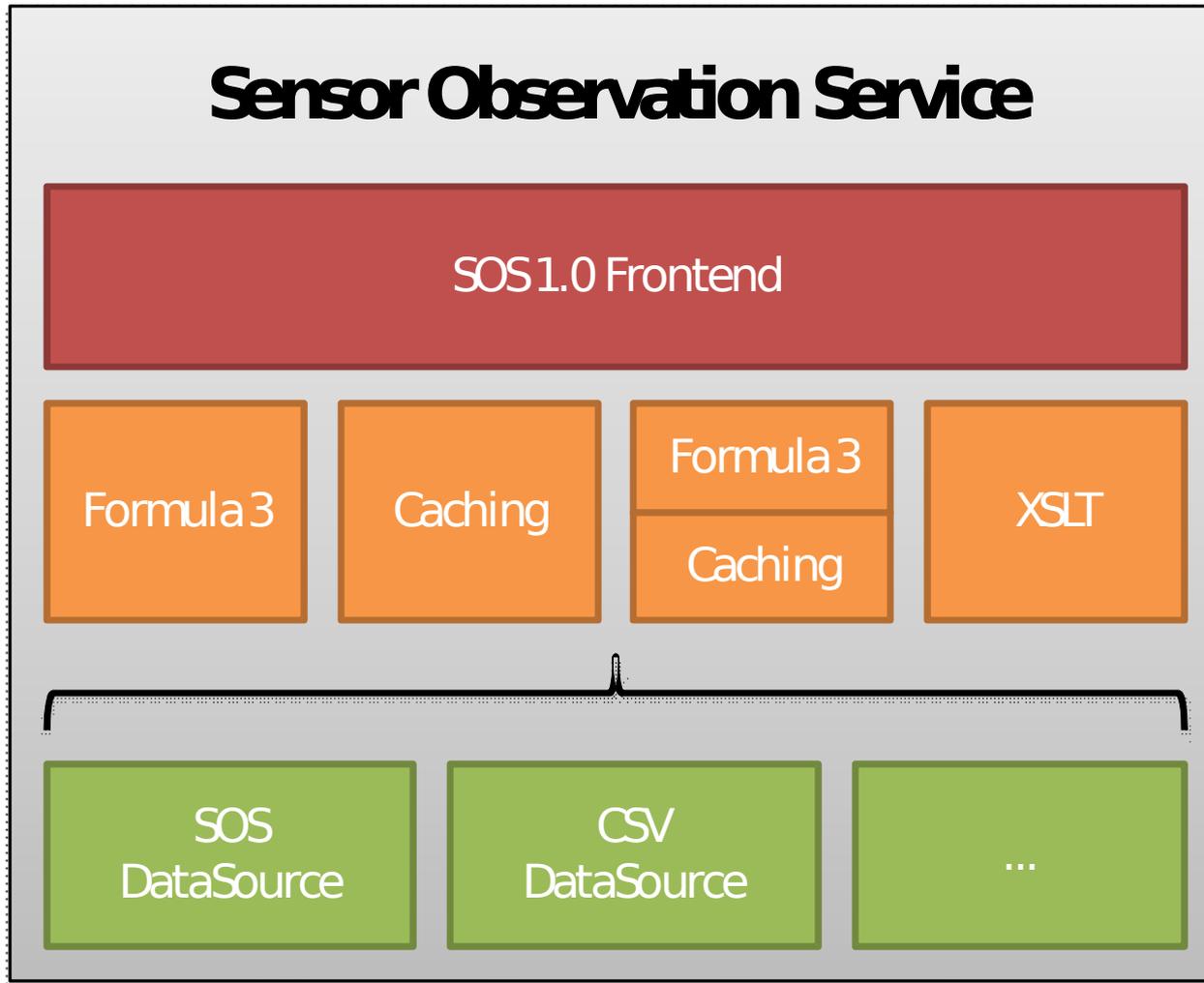
# SOS Frontend: Use Cases

- SOS interface to DataSource (CSV, legacy, etc.)

- SOS with on-the-fly processing

- Cascading SOS: custom views, transformations, caching, etc.

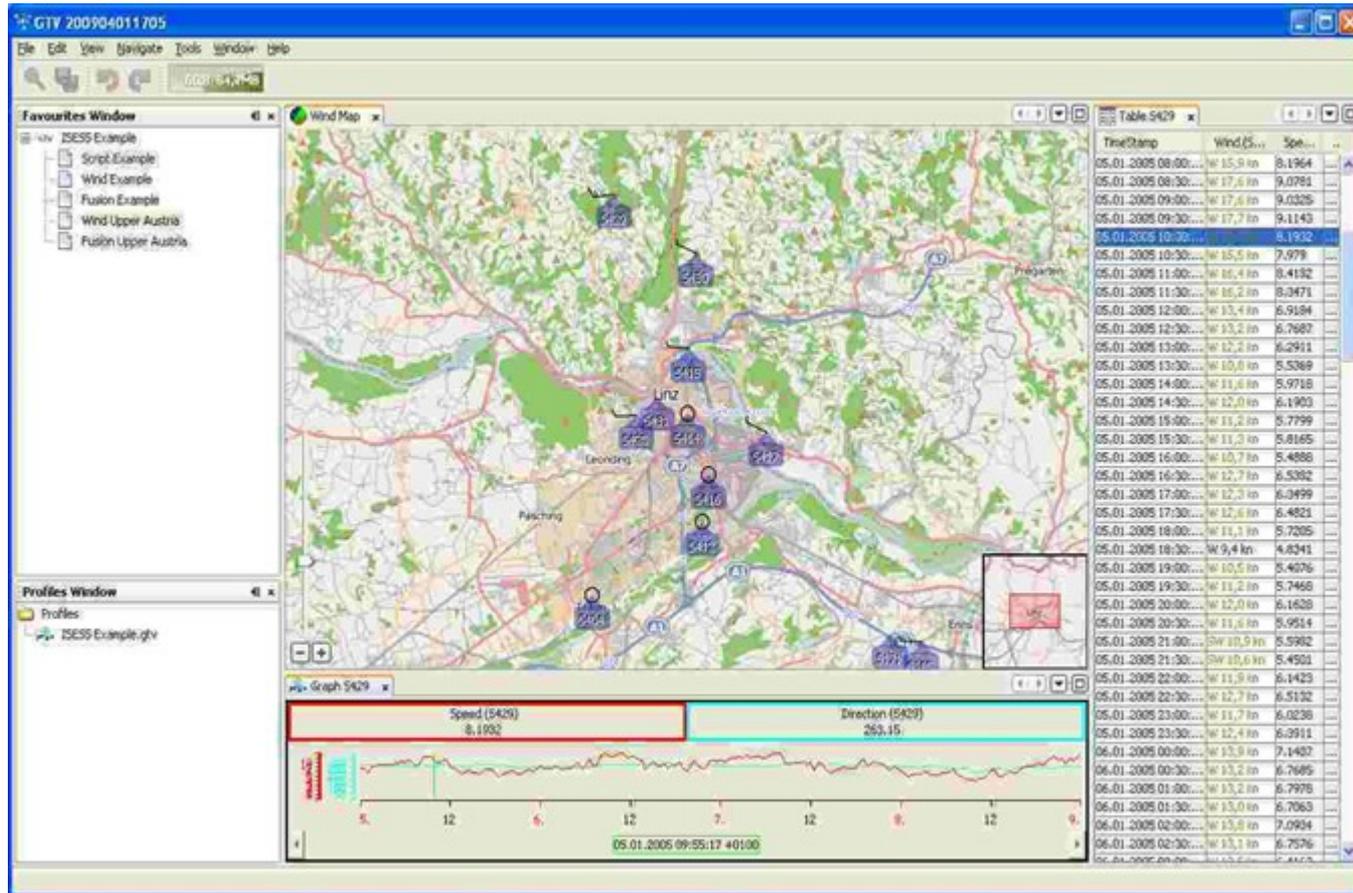- Data preparation for reporting

- ...

# Universal Data Pump

- Read data from a DataSource

- Write to one or more DataSink(s)

- Configurable (config file and commandline)

- Batch processing

- CSV $\rightarrow$ SOS, SOS $\rightarrow$ CSV, Legacy $\rightarrow$ SOS, AnySen $\rightarrow$ SOS, etc.

# Data Pump: Example Scenario

Export data from SOS into a CSV file (e.g. for legacy modeling applications)
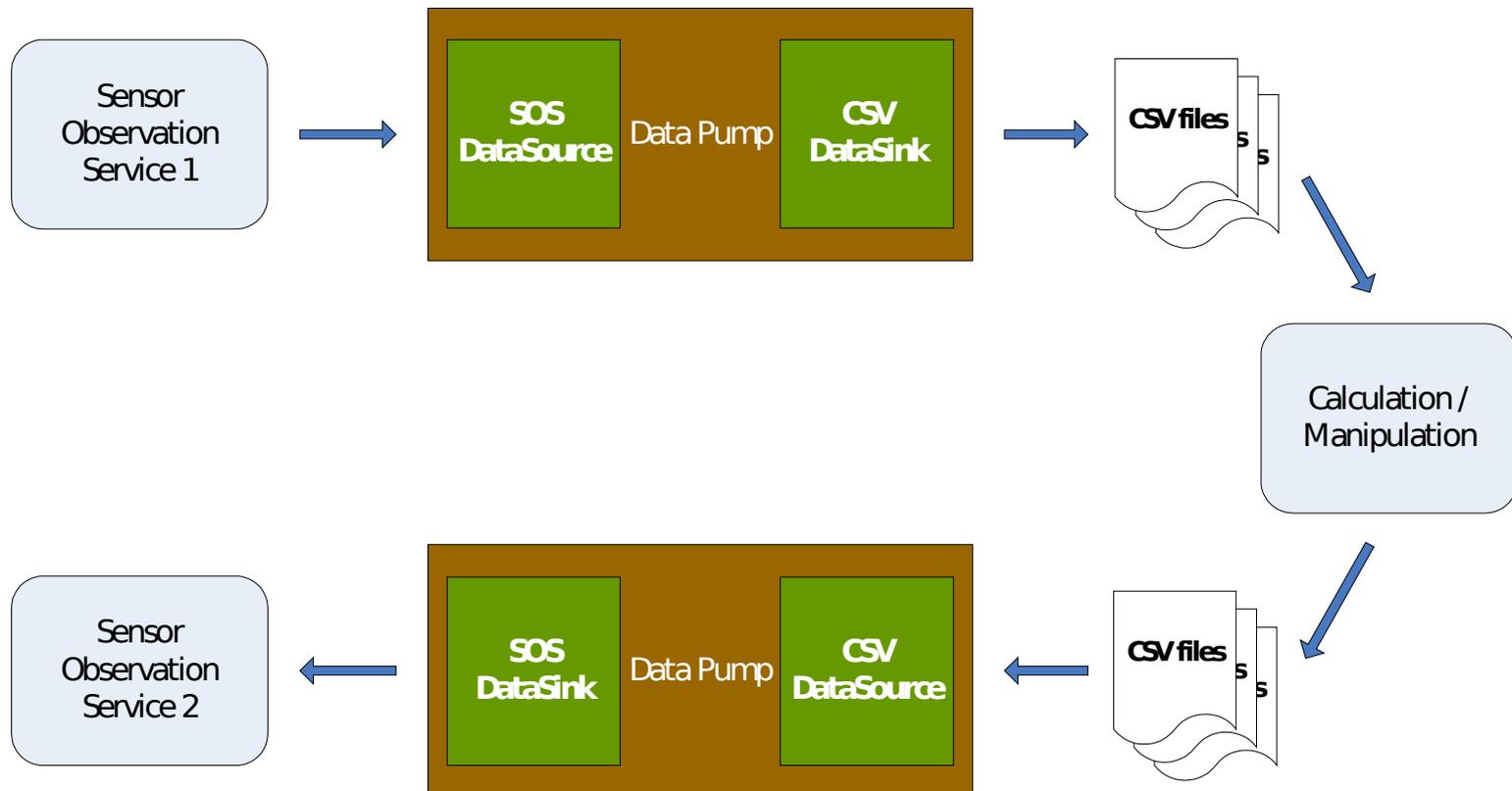
# Georeferenced Timeseries Editor (GTE)

# GTE: What is it?

- A generic desktop application, which allows viewing and editing of georeferenced time series data

- A toolbox for building specialized applications capable of presenting a common and combined view on time series data from different sources

- The main GTE components are the connectors (TS-Toobox data handlers, and processing components), central GTE controller component, and the interactive viewer windows, which can be combined and configured in a flexible way

- Both, connectors and viewers can be easily added at runtime without the need to recompile or reconfigure the application

# Time Series Toolbox Availability

- Developed by AIT within the SANY project

- Available under dual license: Open source (GPL) and commercial license

- Prototype release at: http://sourceforge.net/projects/timeseriestool/

- New full release planned shortly

- For more info contact: bojan.bozic@ait.ac.at

# AIT Austrian Institute of Technology

your ingenious partner

Bojan Božić, MSc

Safety & Security

Information Management and eHealth


AIT Austrian Institute of Technology GmbH

4333 Seibersdorf | Austria

T +43(0) 50550-3190 | M +43(0) 664 2351 800

bojan.bozic@ait.ac.at | http://www.ait.ac.at