PermaSense Data Management

System documentation and tutorial for online data access

April 9, 2014

Samuel Weber, Jan Beutel, Christoph Walser

PermaSense

THE PERMASENSE SYSTEM

This section gives an overview about:

- The PermaSense system architecture
- GSN-based data management:
  - data flow
  - servers and virtual sensors
  - web user interface
  - timing information

PermaSense

PermaSense System Architecture

Sensor Node (SN)

- Collects data from different sensor options

Wireless Sensor Network

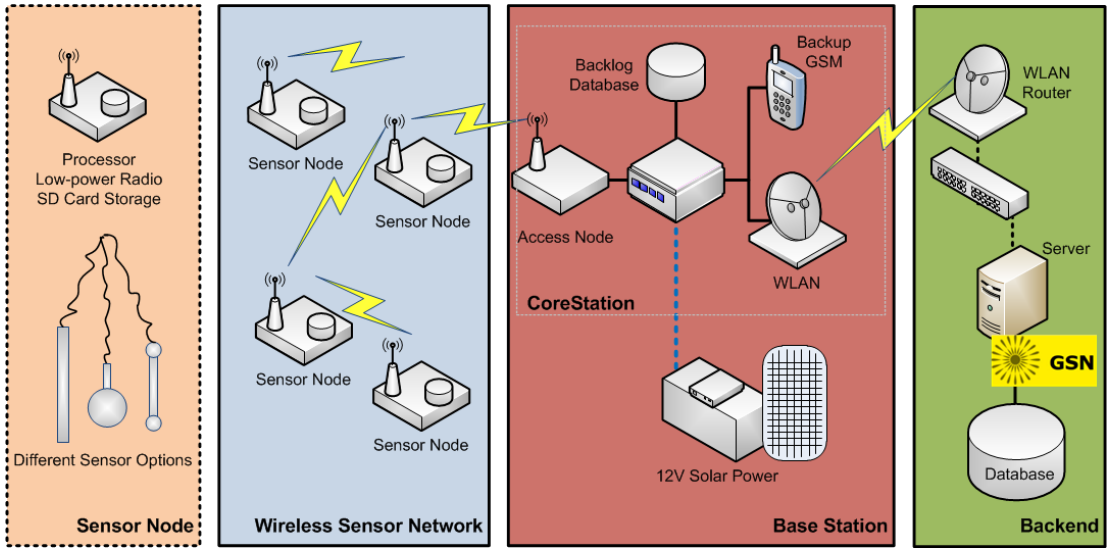
- Forwards the collected data over a 868 MHz wireless communication channel to the base station

Base station

- Sinks all data and forwards it to a central server over an IP network

Backend- and GSN-Server

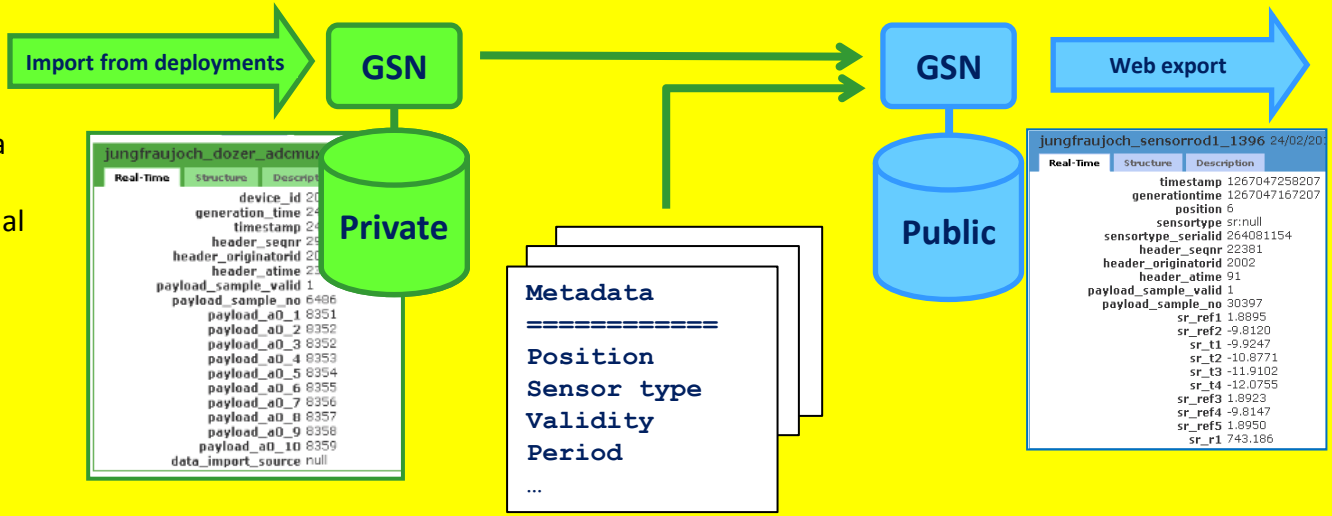
- Collects data from all deployments and implements a number of management and monitoring services



PermaSense GSN Architecture: Hierarchical Online Data Processing

Global Sensor Network (GSN)

- Data streaming framework from EPFL
- Organized in “virtual sensors”, i.e. data types/semantics
- Hierarchies and concatenation of virtual sensors enable on-line processing
- Dual architecture translates data from machine representation to SI values, adds metadata

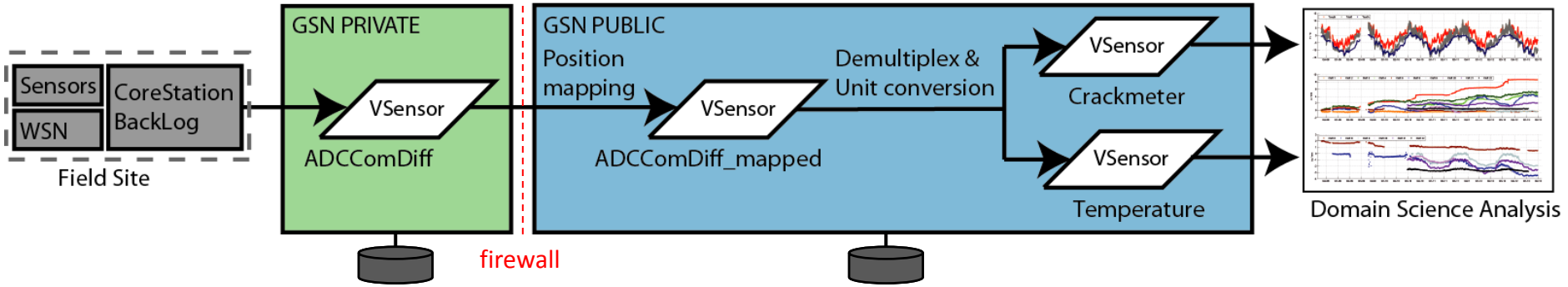


GSN Server Data Flow

Data from field site is received by the private GSN server “as is” and stored in a primary database.

Data is passed on to a public GSN server where it is mapped to positions, sensor types and converted to convenient data formats.

Data is downloaded, analyzed and plotted using external tools.



- All data from sensor nodes to backend is transmitted in packets

  The length of the sensor network data packets is limited to 23 bytes. In cases where sensor data is larger, several consecutive packets are generated.

- Data is received and sorted according to data types

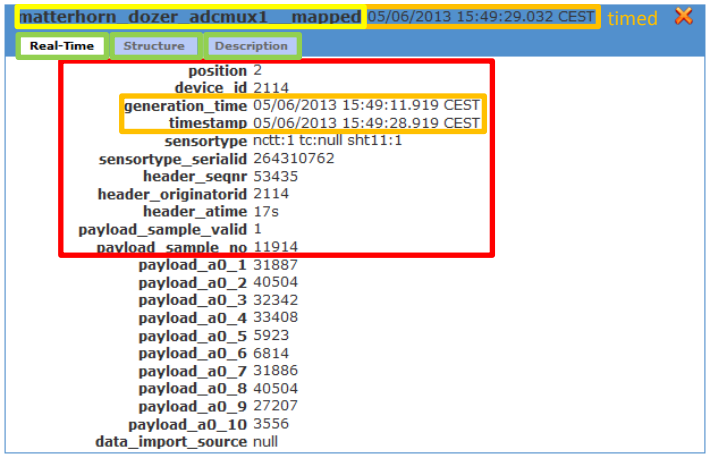
PermaSense

Structure of a Virtual Sensor

- Data in database (GSN) is organized as virtual sensors (VS) per deployment
  If there are multiple sensors yielding the same data types, this data is multiplexed into the same VS.

- Each VS has a unique name: *<deployment>_<sensor type>_<processing step>*
  Processing steps: *raw* → *mapped* → *converted*

- Each VS has three tabs with different information



- Each VS contains a header. The header for sensor network specific data contains:
  - *position*: number of physical locations
  - *device_id*: mapped device id at this position
  - *timed/generation_time/timestamp*: there are several time formats that are described later
  - *sensortype*: sensor types and calibration constants connected at a given position
  - *sensortype_serialid*:
  - *header_seqnr*: sequence number denoting successively generated sensor network packets
  - *header_originatorid*: the same as device id
  - *header_atime*: packet transfer time in seconds, used to calculate generation time
  - *payload_sample_valid*: flag which specifies data acquisition errors
  - *payload_sample_no*: sample number denoting packets originating from same data sampling period (typ. period is 120 sec)

PermaSense

GSN Web User Interface

http://data.permasense.ch

HOME

All virtual sensors are listed and sorted by deployment, data type, sensor type and process step

DATA

Manual download of data

NETWORK TOPOLOGY

Topology and table with network information sorted by deployment

LOGS

Log files from core stations

BACKEND

Backend, mySQL and Zabbix monitoring server specific plots

SENSOR NETWORK

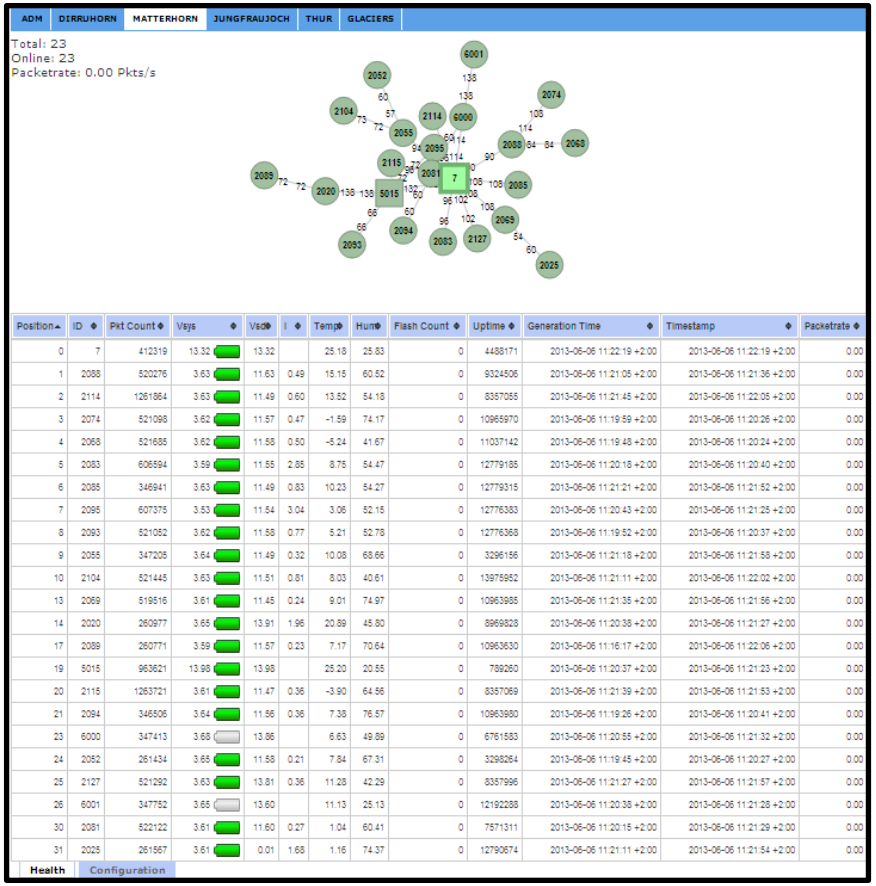
Sensor network specific plots

SCIENCE

Collection of scientific plots

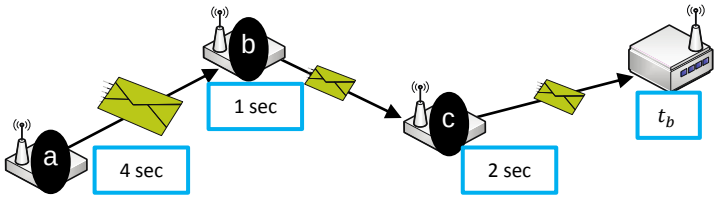
ON-SITE WEATHER

Weather information and webcams



GSN Multiple Time Bases

- Global reference time (UTC) is often not available
  - → Solution: Elapsed time on arrival
    - Sensor nodes measure/accumulate sojourn time
    - Base station annotates data with arrival time (e.g. UTC)
    - Generation time is calculated as difference between arrival time and sojourn time



$\tilde{t}_b = t_b - \tilde{t}_b$ =2013/06/06 17:47:11 – (4+1+2)=2013/06/06 17:47:04

- All data carries multiple timestamps
  - *generation_time* depicts time when data is sampled
  - *timestamp* denotes the time when data reaches a UTC synchronized time base for the first time (e.g. base station)
  - *timed* is database time, i.e. the time data is inserted into the database

- GSN time
  - Is in UTC (Switzerland: winter +1h (CET), summer +2h (CEST))
  - Is unix-time in millisecond!
    - *Unix time stamp* is merely the number of *seconds* between a particular date and the Unix Epoch (January 1, 1970).
    - *GSN time stamp* is merely the number of *milliseconds* between a particular date and the Unix Epoch.
  - Example for time conversion in Matlab:

```
function mt=time_gsn2matlab(gt)
% converts times in gsn format to time in matlab datenum
% format, UTC time GMT+2
mt=datenum(1970,1,1) +gt./(3600*24*1000) +datenum(0,0,0,2,0,0);
return;
```


PermaSense

Metadata for Deployments

- PermaSense maintains four deployments,

  Matterhorn (MH), Jungfraujoch (JJ), Dirruhorn (DH), Aiguille du Midi (ADM)
- Metadata for each deployment is described in the *nodeposition.xls* file
  - Name of the deployment
  - Position: physical location of a sensor type (shown in an additional map or overview picture)
  - Sensor type used at the position
  - Coordinates of the position
  - Device ID: unique ID identifying a piece of hardware

Deployment Overview: Aiguille Du Midi

green = new installation | radiochannel = 12

										Date	31.05.2013	06.05.2013	08.07.2012
										# Devices	18	17	12
Position	Label	Name	Comment	Host Name	Sensor ID	Sensor Type	CR Length	GPS X	GPS Y	GPS Alt			
0		Base Station		permasense-caa-adm-1				6.887489	45.879446	3750	29	29	29
1		West Face Piton Central Top Crack			SN 10270716	NCTN	100	6.88762	45.878993	3750	2121	2121	2121
2		West Face Piton Central Bottom Crack				NCNN	150	6.887391	45.879083	3750	2105	2105	2105
3		West Face Piton Nord				NCNN	200	6.886823	45.879243	3750	2110	2110	2110
4		South Face Piton Nord				NCTN	150	6.886902	45.879011	3750	2109	2109	2109
5		East Face Piton Nord				NCNN	100	6.887519	45.87931	3750	2108	2108	2108
6		North Face Piton Nord				NCTN	200	6.887177	45.879532	3750	2111	2111	2111
7		Borehole South				CR1000		6.887523	45.878458	3750	2131		
8		Borehole North				CR1000		6.886813	45.878913	3750	2130	2130	
9		BoreHole East				CR1000		6.888173	45.878611	3754	2129	2129	
10		Relay South Piton Central						6.887625	45.878244	3741	2120	2120	2120
11		Argentière Glacier Webcam				Webcam					6000	6000	6000
12		Aiguille du Midi Webcam				Webcam					6001	6001	6001
13		Arrete Aiguille du Midi Webcam				Webcam					6002	6002	6002
14		Mont Blanc Webcam				Webcam					6003	6003	6003
15		Relay North Top Piton Central						6.887902	45.878943	3825	2066	2066	
16		Relay West Terrace Piton Nord						6.886505	45.878946	3723	2054	2054	
17		Relay Cable Car						6.887404	45.879565	3710	2057	2057	

- If an additional sensor gets installed, a new position is created
- If a sensor gets replaced, a new data column is added with the new device ID

PermaSense

DATA ACCESS

This sections gives an overview about data access using external tools.

Accessing Data from the GSN Server

- There are two different methods to get data from GSN using *http queries*:
  1. ***One-shot query approach:*** each request returns data based on the database state at the time of the query. This allows one to quickly obtain aggregated data from a virtual sensor and export this data in convenient formats, e.g. CSV or XML.
  2. ***Streaming approach:*** requested data is continuously streamed to the user in real-time until the connection to the server is closed.

- Example for a simple *one-shot query* without aggregation/conditions
  - Suppose, you want to query all fields of the *matterhorn_crackmeter__tctc* virtual sensor (2 crackmeters, 2 thermistors) between 25/08/2012 and 13/06/2013 (UTC):
    1. Open in browser:
       http://data.permasense.ch/multidata?vs[0]=matterhorn_crackmeter__tctc&time_format=iso&field[0]=All&from=25/08/2012+00:00:00&to=13/06/2013+00:00:00

       where: vs[0]: name of virtual sensor
       time_format: time format of returned data
       field[0]: list of data fields to return
       from, to: time limits of data request in UTC

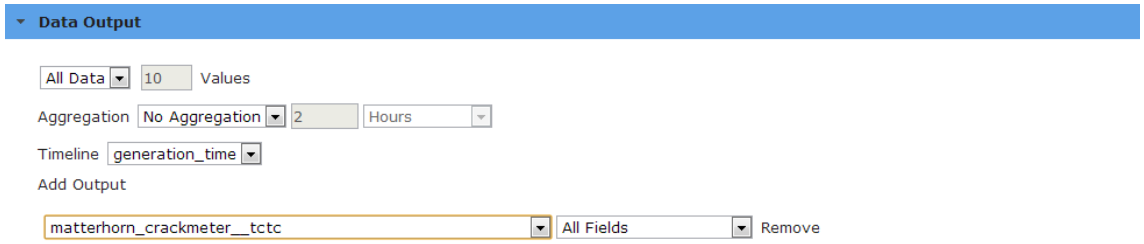
       For further options and syntax information please consult:
       https://github.com/LSIR/gsn/wiki/Web-Interface#multidata
    2. You will get a CSV-formatted file with the requested data.
  - A complete example using Matlab can be downloaded

PermaSense

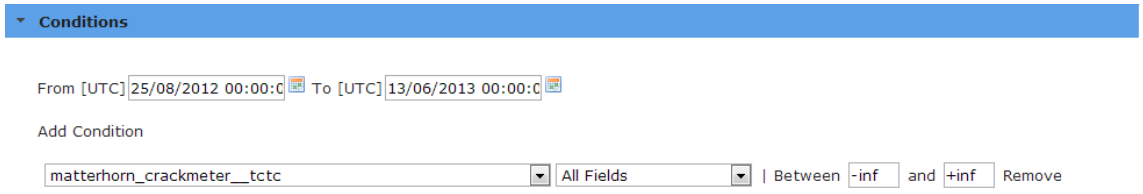
Accessing Data over the *data.permasense.ch* Web Interface

http://data.permasense.ch

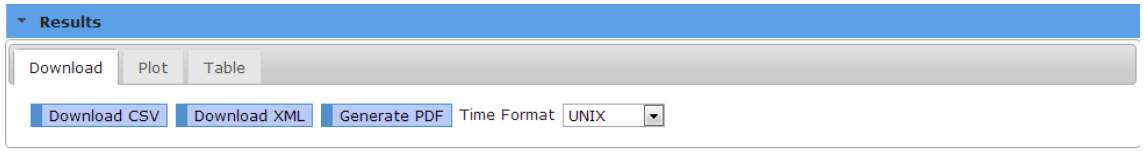
- Manual access to all virtual sensors and data fields over web interface
- Example (same as for one-shot query):
  - Get all fields of the *matterhorn_crackmeter__tctc* VS between 25/08/2012 and 13/06/2013:
    1. Go to http://data.permasense.ch, open tab '*DATA*'
    2. In sub-tab '*Data Output*', select '*All Data*' from the first drop down menu and the according virtual sensor:



    3. Click on sub-tab '*Conditions*' and set the time limits:



    4. Click on sub-tab '*Results*' and access the data in one of the offered formats:



Note: The plot is interactive and can be used to easily preview the resulting data.

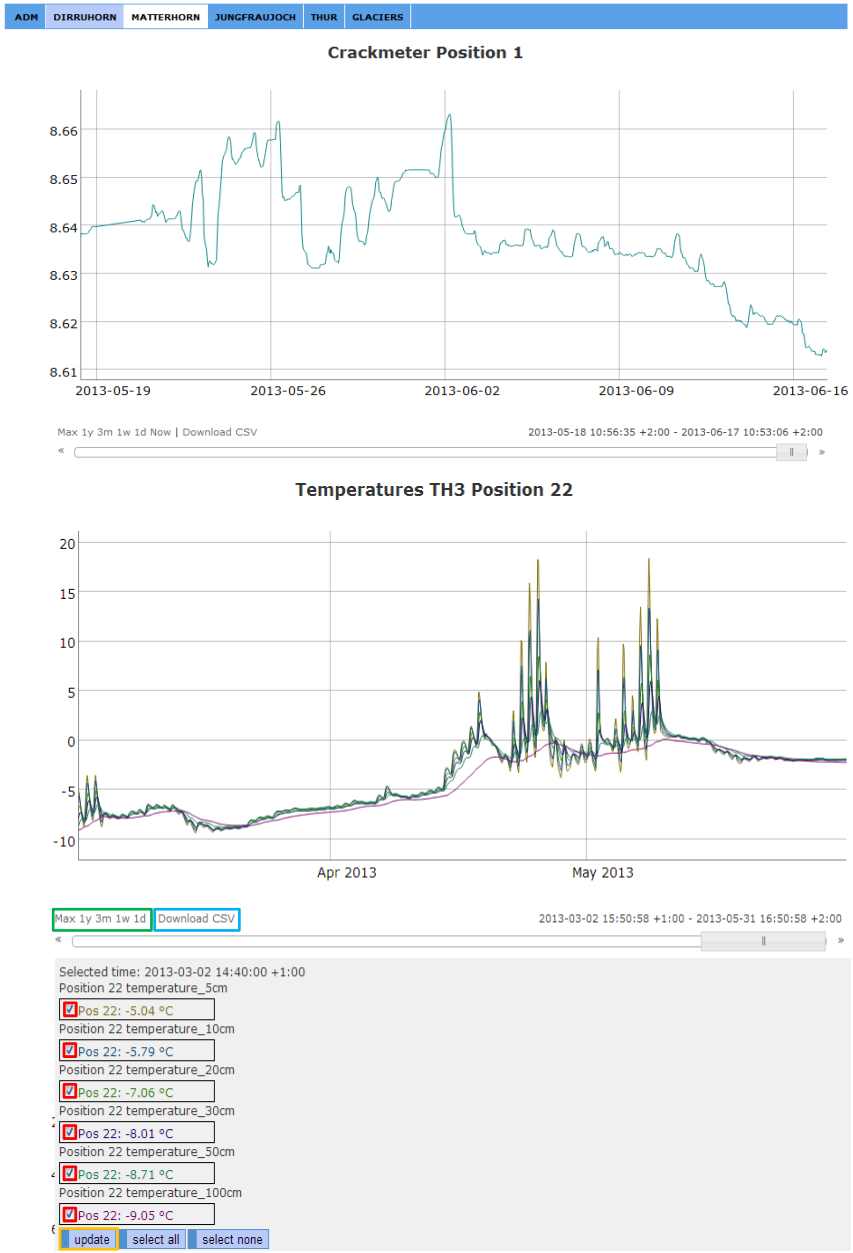
PermaSense

Accessing Data over the *data.permasense.ch* Web Interface

- A selection of predefined plots can be found in the tabs *SCIENCE* and *SENSOR NETWORK*

- Plots are generated using the *Vizzly* framework

- All plots are interactive:
  - Time: time range to be shown
  - Pos: data fields to be plotted
  - *Update:* has to be clicked after ticking
  - *Download CSV* download aggregated of currently shown plot



PermaSense

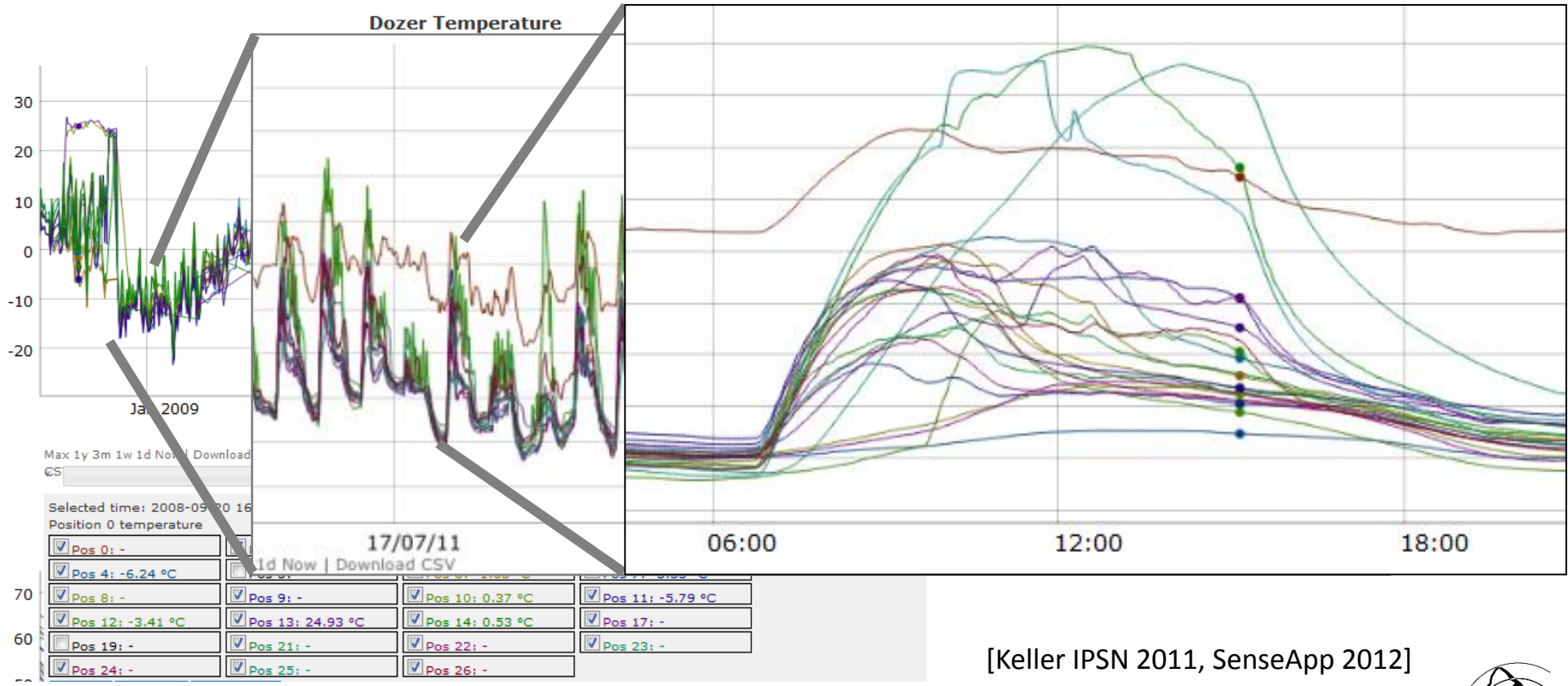
BACKGROUND ON GSN AND SYSTEM ARCHITECTURE

This sections gives an overview about:

PermaSense

Vizzly: Visualization of Large Data

- Fast access to millions of data samples
- Pan, zoom, channel selection
- Combination of historic and real-time data



[Keller IPSN 2011, SenseApp 2012]

PermaSense

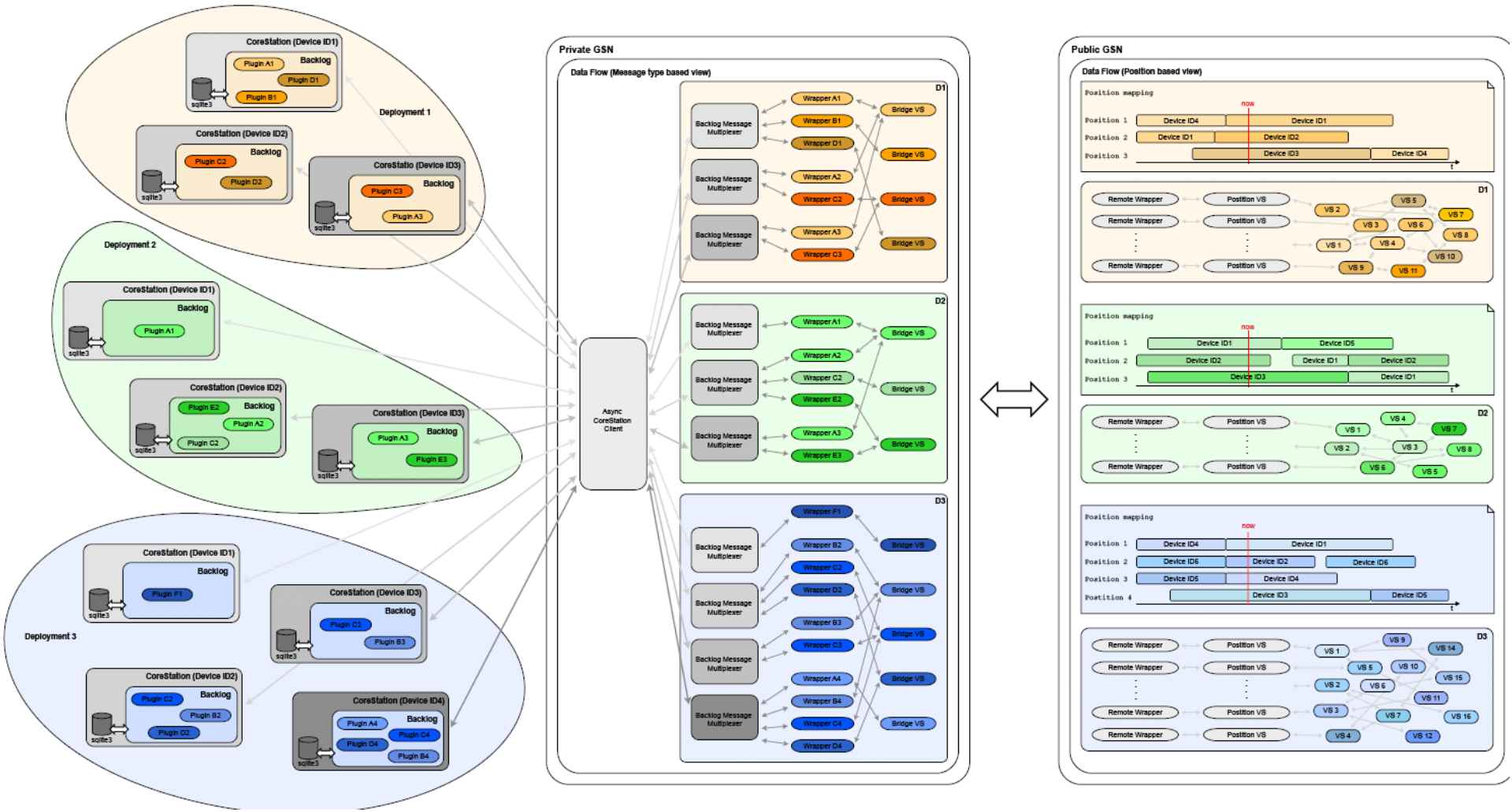
WSN On-Node Storage Layer

- On-node flash based storage (SD-Card)
  - Integrated with Dozer queuing mechanism (beacon traces & per-link ack's with backpressure)
  - All generated packets are stored on local flash memory
  - Packets not yet sent are flagged for sending later
  - Bulk access optimized for flash memory (no single packet transfers)
- Enables both delayed sending (disruptions) and post-deployment validation

Mitigating Post WSN Data Loss

- BackLog = Auxiliary data aggregation layer at device level
  - Remote storage and synchronization layer for Linux systems
  - Python based, designed for PermaSense CoreStation
  - Plugin architecture for extension to custom data sources
  - Data multiplex from plugin to GSN wrapper over one socket
- Reliable (flow controlled) synchronization
- Schedulable plugin/script execution, remote controlled

Multi-Site, Multi-Station, Multi-Revision Data…



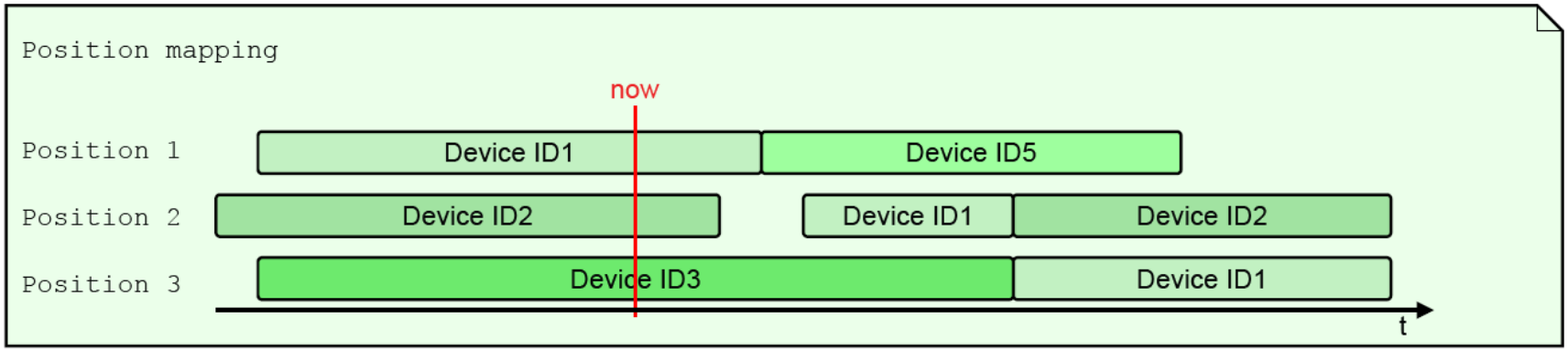
PermaSense

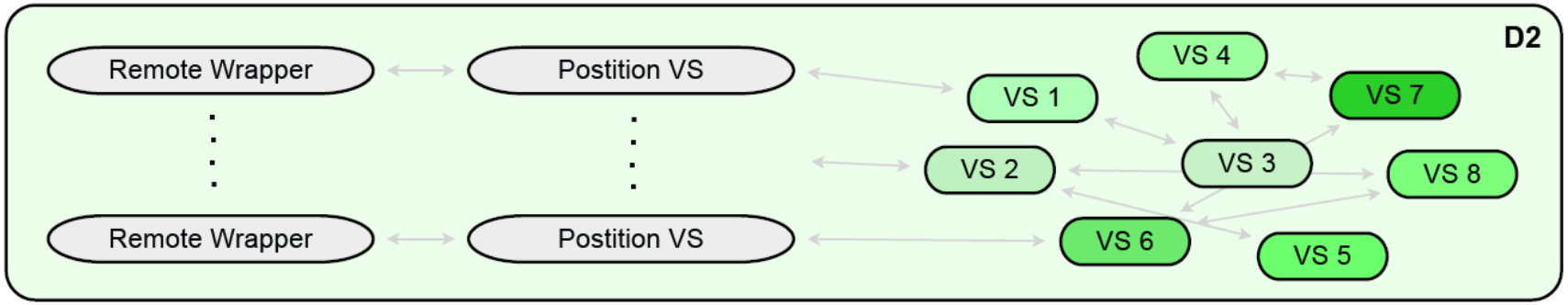
Metadata Mapping Architecture

- Based on 2 GSN instances
  - Separation of load/concern across two machines
  - “Private” GSN instance, raw data, protected, high availability
  - “Public” GSN instance, mapped and converted data, open, non-critical
- Metadata stored in version control system (CSV, SVN)
- Mapping of
  - Positions, coordinates, sensor types, conversion functions, sensor calibration…
- Conversion of
  - Time formats, raw to SI values…
- Replay of metadata/mapping possible, e.g. on errors
- Change management

Metadata Change Management

- Allows simple exchange of sensor hard-/software at runtime
- Post-deployment annotation
  - Stop GSN– deployment change – annotate metadata – restart GSN
- Automatic synchronization with 1 day change boundaries





PermaSense