Integration of different data sources into 52N SOS
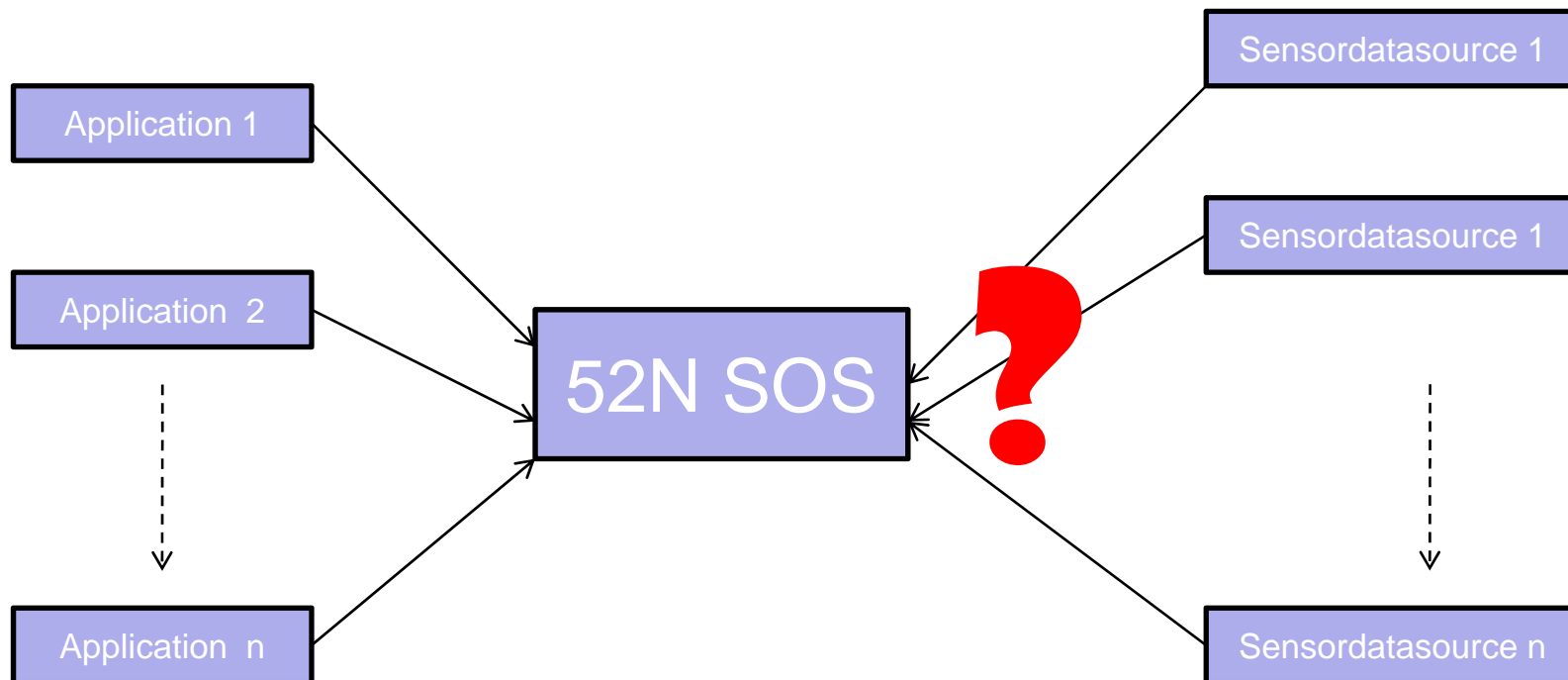
52north
exploring horizons

Open Source SDI Technologies

Christoph Stasch

52° North SOS

- Conform to SOS specification: *06-009r5*

- Supports Core and Transactional Profile + GetResult, GetFeatureOfInterest, GetObservationByID

- DBMS: PostgreSQL + PostGIS

- customized implementations:
    - OracleSpatial
    - ArcSDE
    - …
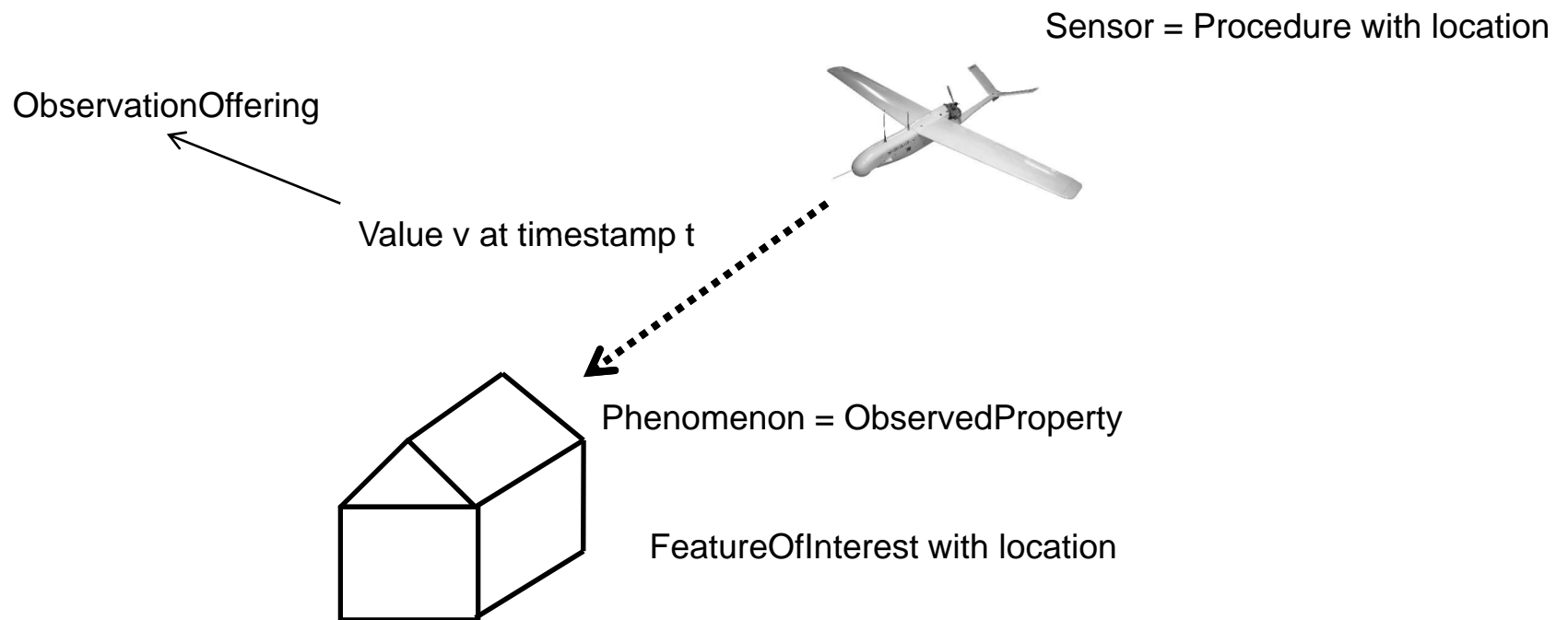
- Mobile extension

Motivation

Agenda

- Modelling your data

- 52°North implementation of SOS-T

- 52°North SOS Feeder Framework
  - Overview
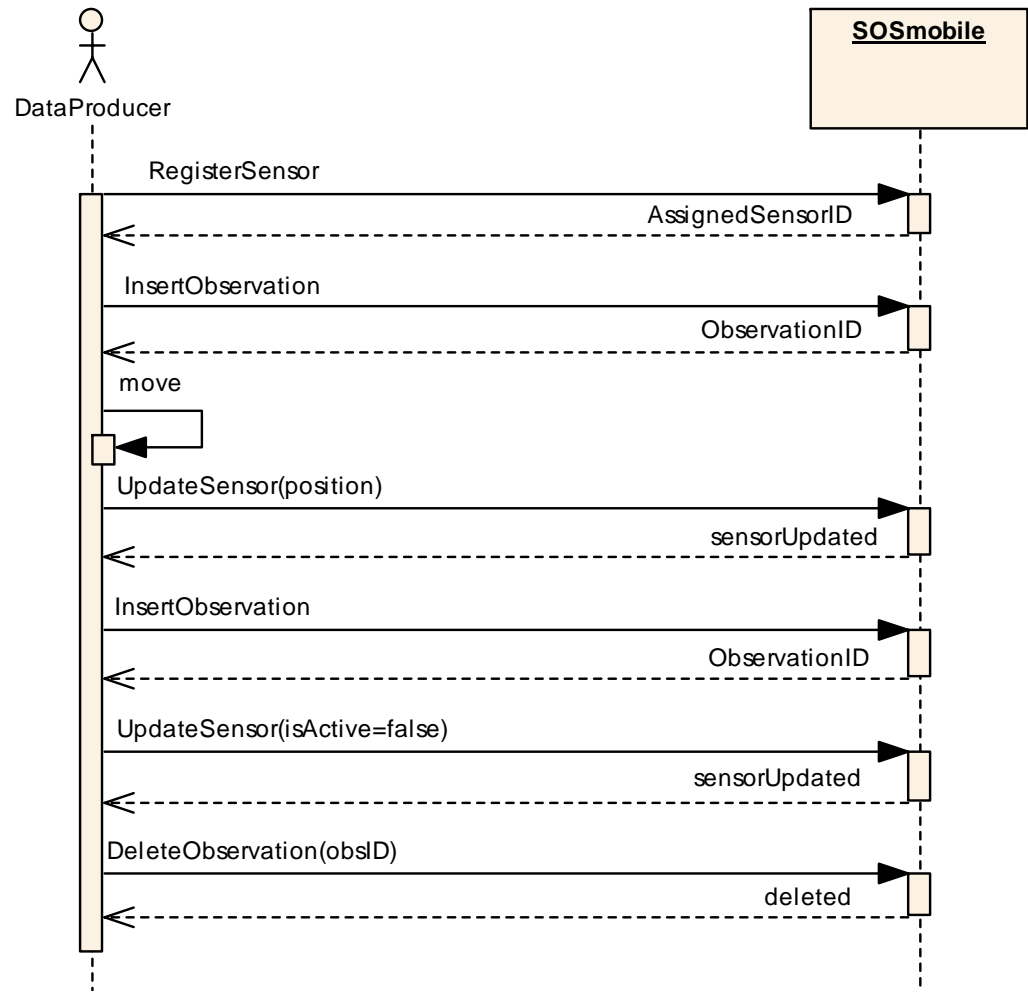 - Architecture
  - Installation

Modelling your data

- **Identify O&M components in your data**

Sensor = Procedure with location

ObservationOffering

Value v at timestamp t

Phenomenon = ObservedProperty

FeatureOfInterest with location

- **O&M components are mapped onto 52°North SOS Database Model**

52°North SOS-Transactional

DataProducer

SOSmobile

RegisterSensor

AssignedSensorID

InsertObservation

ObservationID

move

UpdateSensor(position)

sensorUpdated

InsertObservation

ObservationID

UpdateSensor(isActive=false)

sensorUpdated

DeleteObservation(obsID)

deleted

RegisterSensor Operation

- Register new sensors at SOS

- Parameters:

  – Sensor Description (smI:System)

    - ID

    - ObservationOffering

    - Output phenomena of sensor system

    - Position

    - Mobile

    - Active

- Returns assigned Sensor ID

InsertObservation

- Insertion of new measurements into SOS
- Currently only numerical observations supported (om:Measurement)
- Parameters:
  - Sensor ID
  - Measurement
- Returns ObservationID

UpdateSensor Operation

- Extension of 52°North implementation

- Updating of dynamic metadata

- Parameters
  - ID
  - Position
  - Mobile
  - Active

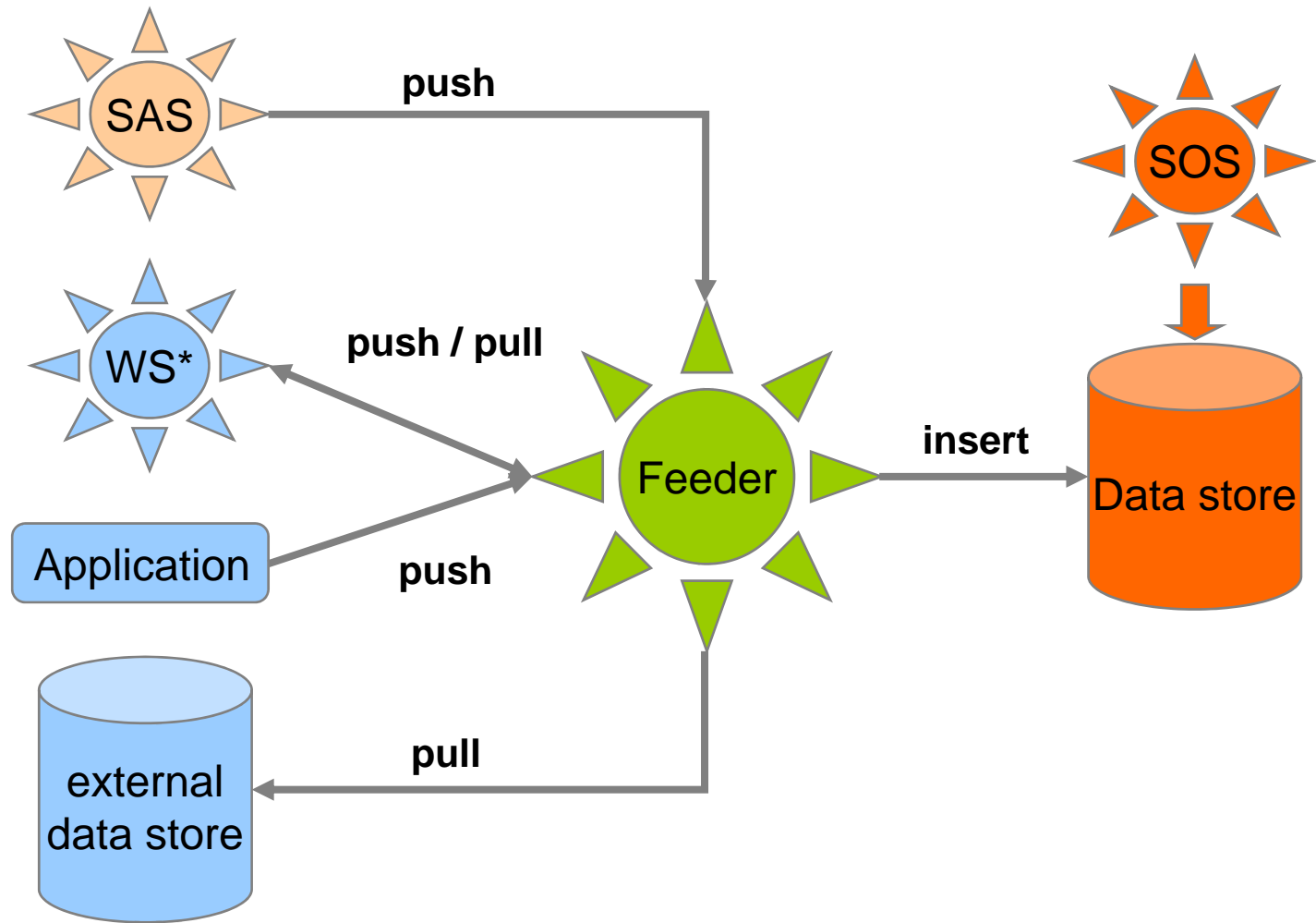- Returns information, whether update was successfull

DeleteObservation Operation

- Extension of 52°North implementation
- Deleting of observations
- Parameters
  - Observation ID
- Returns information, whether update was successfull

52° North SOS Feeder Framework

- Web application for PUSH and PULL of sensor data
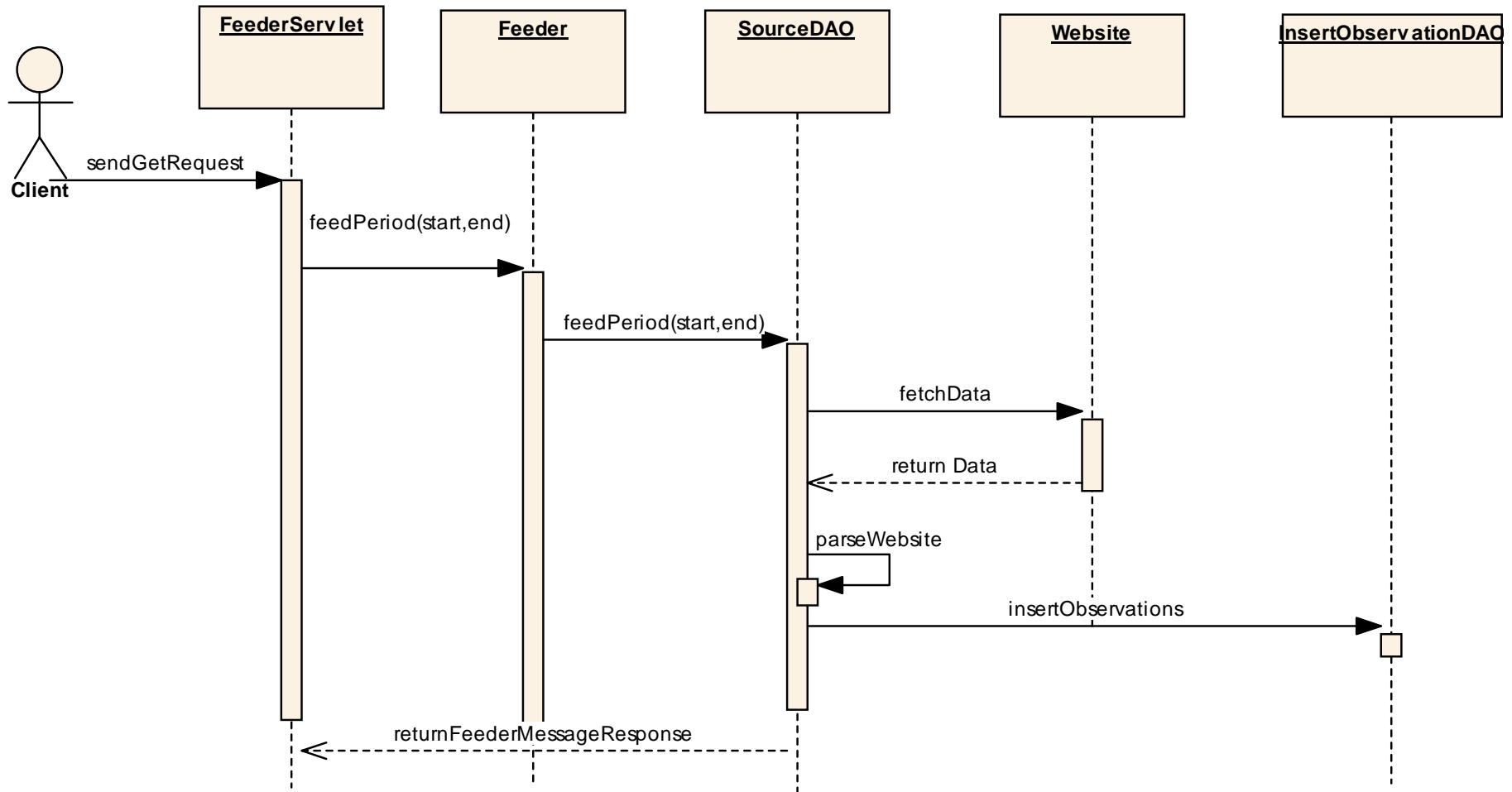- DAO implementations for SOS Database

52° North SOS Feeder Framework

SAS —**push**→

WS* ←**push / pull**→

Application —**push**→

Feeder —**insert**→ Data store

SOS → Data store
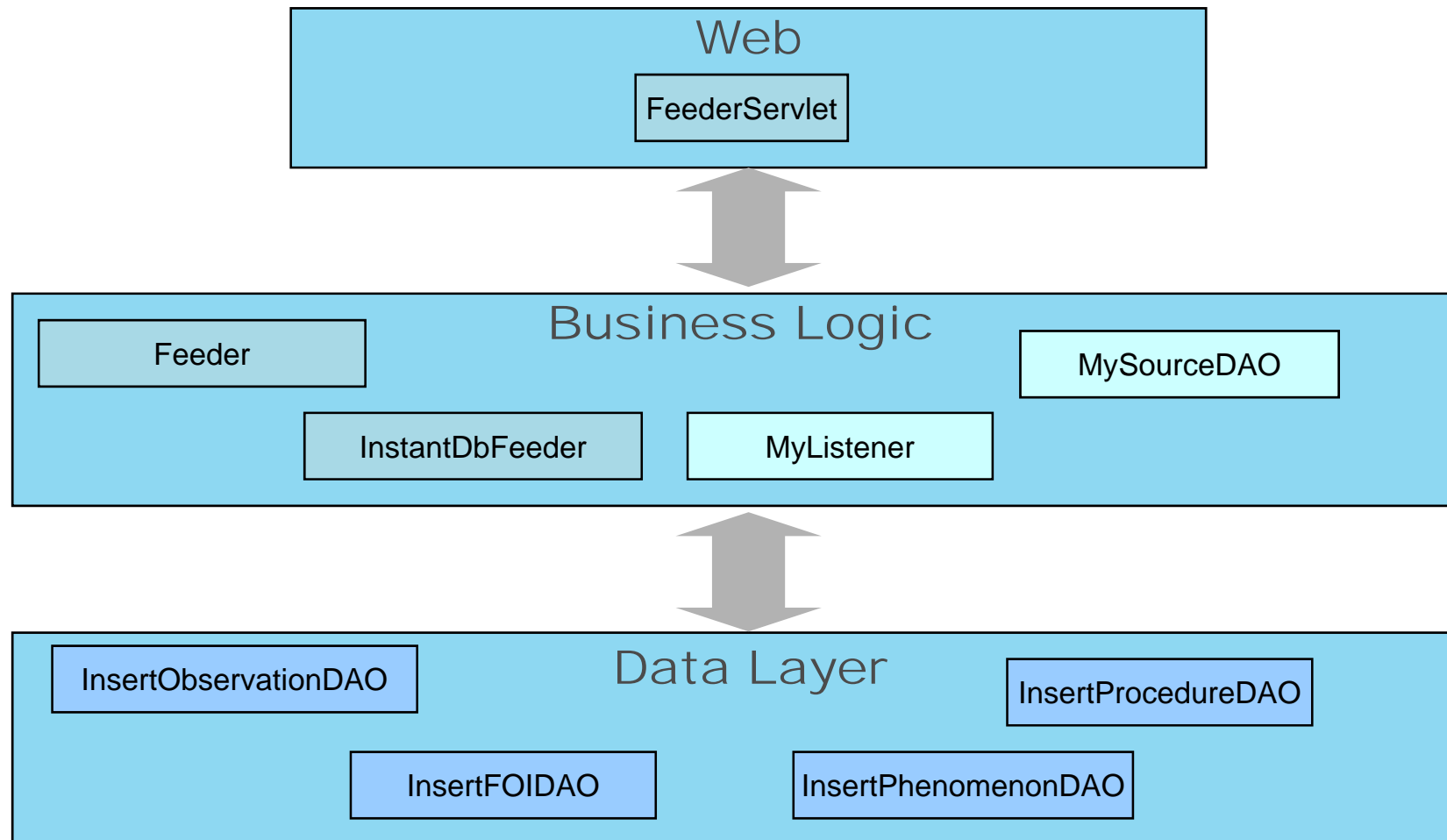
external data store ←**pull**—

Operations

- HTTP GET for PULL of data from source into DB:
  - InitializeDatabase
  - FeedPeriod with parameters Begin and End
  - StartInstantFeeding
  - StopInstantFeeding
- HTTP POST for PUSH of data into DB:
  - XmlBean is created from XmlFile, which is handed on to the SosFeeder

PULL sequence

3-Tier Web Architecture

Web

FeederServlet

Business Logic

Feeder

MySourceDAO

InstantDbFeeder

MyListener

Data Layer

InsertObservationDAO

InsertProcedureDAO

InsertFOIDAO

InsertPhenomenonDAO

Implement SosFeeder (PULL)

- Implement `AbstractSourceDAO`
  - Use InsertDAOs for easy insert of data into SOS database
- Set name of implementation of `AbstractSourceDAO` in `dbfeeder.config` file
- Add additional (optional) properties
- Deploy feeder and send different requests:
  - http://yourUrl/feeder?REQUEST=InitializeDatabase
  - http://yourUrl/feeder?REQUEST=StartInstantFeeding
  - http://yourUrl/feeder?REQUEST=FeedPeriod&BEGIN=01.01.2007&END=31.01.2007

SourceDAO

- Needs to be implemented for PULL of data from datasource into database
- Contained in package `org.n52.sos.feeder.source`
- `fetchData()` is invoked when periodically fetching data
- `fetchPeriod()` should fetch data for the passed time interval
- `initializeDatabase()` should insert (static) metadata for observations (e.g. offering, phenomena, etc.)
- Name of class must be set in `dbfeeder.config` file

AbstractSourceDAO
+ *fetchData() : void*
+ *fetchPeriod(begin,end)() : void*
+ *initializeDatabase() : void*

SosFeeder – periodic feeding

- SosFeeder offers possibility to fetch Data periodically (e.g. every 10 minutes) from a datasource
- Class `org.n52.sos.feeder.source.AbstractSourceDAO` needs to be implemented
- `InstantFeederTask` for inserting data; invokes `fetchData()` method of `AbstractSourceDAO` implementation
- Period (in milliseconds) is defined in `feeder.config` file

That's it!

- Further information available at:

    https://incubator.52north.org/twiki/bin/view/ Sensornet/SensorObservationService

    or ask directly at: swe@52north.org