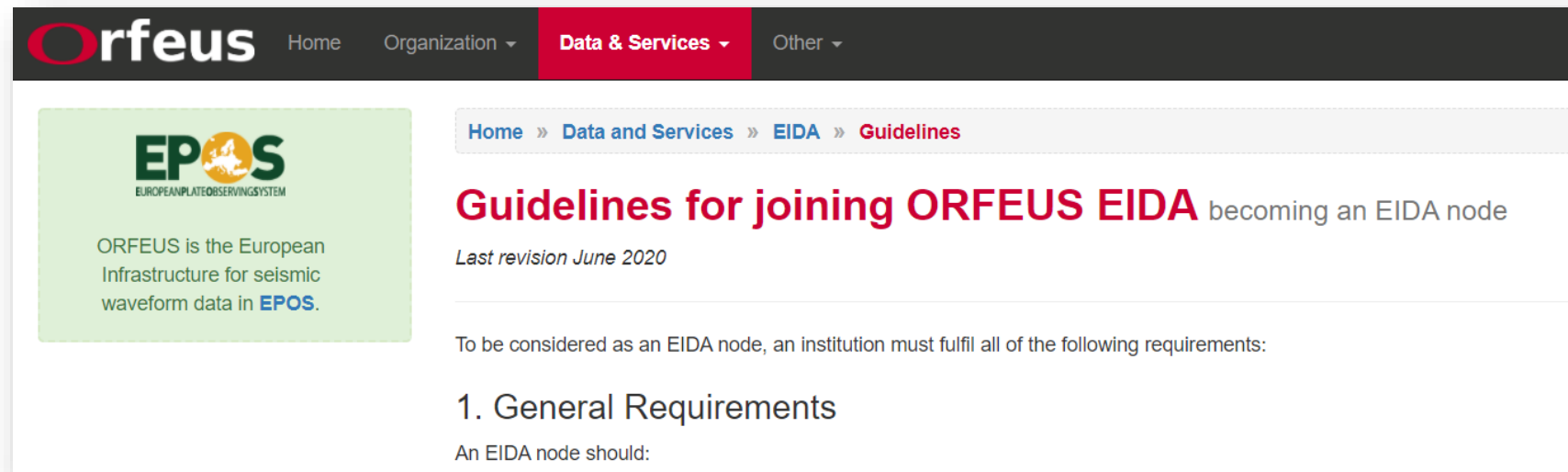# EIDA knowledge transfer

The joint 2020 Nordic Seismology and Nordic EPOS Seminar

30 Sep – 2 Oct 2020

Jan Michalek

Outline

- Getting ready for **European Integrated Data Archive** (EIDA)
  - Technically
  - Formally: https://www.orfeus-eu.org/data/eida/guidelines/

- Norwegian EIDA node

Getting ready for EIDA

Getting ready for EIDA – Technically (1)

- Most EIDA nodes are running **SEISCOMP** which provides
  - Data collection + archive
  - **FDSN web services: DATASELECT, STATION**
  + other modules

  - Download: https://github.com/SeisComP/seiscomp
  - Installation: https://www.seiscomp.de/doc/base/installation.html
  - Configuration: https://www.seiscomp.de/doc/base/getting-started.html#getting-started

  - **fdsnws** server configuration

Getting ready for EIDA – Technically (2)

- Other services are required by EIDA
  - EIDA web services
    - WFCATALOG: provides metrics about waveform data quality and availability
      - Download: https://github.com/EIDA/wfcatalog
      - Installation: https://dev.knmi.nl/projects/eida/wiki/WFCatalog
      - Consists of two parts – *collector* (calculates metrics; cronjob) and *service* (expose data via web service)

    - ROUTING: provides information about services at EIDA nodes
      - Download: https://github.com/EIDA/routing
      - Installation and configuration: https://github.com/EIDA/routing/blob/master/doc/userdoc.rst

Getting ready for EIDA – Technically (3)

- **Other tools**
  - **Web portal for accessing data**
    - WEBDC3: provides web user interface for interactive data selection
      - Download: https://github.com/EIDA/webdc3
      - Installation: https://github.com/EIDA/webdc3/blob/master/doc/operator.rst
      - It is using FDSN web services, tightly integrated with SEISCOMP

  - **Statistics about data usage**
    - WEBREQLOG: send statistics about FDSN data requests to GEOFON (and other emails)
      - Download: https://github.com/EIDA/webreqlog
      - Installation and configuration: Python script run as cronjob

Getting ready for EIDA – Technically (4)

- Authentication and authorization for DATASELECT
  - Allows access to restricted data to specified users
    - Documentation: https://dev.knmi.nl/projects/eida/wiki/FDSN_-_dataselect_-_auth
    - It is using *arclink-access* binding in SEISCOMP where users can be defined

Getting ready for EIDA – Formally

1. General Requirements (extract from [EIDA guidelines](#))

An EIDA node should:

- be an institution from a **member country of ORFEUS**.
- be committed to an **open data policy**.
- ideally **represent the seismological community at national level or beyond**. In practice this means they should serve as a data repository for other seismic networks in the region.
- commit human resources to contribute to the maintenance and development of the system and associated software.
- commit to active participation in the EIDA Management Board and the EIDA Technical Committee. Note: the additional workload to run an EIDA node as compared to an independent data center is by most nodes evaluated to approximately 0.3 FTE.
- meet a minimum technical level as defined in the ETC Technical Requirements for an EIDA Node. This includes requirements on both data quality (waveform and metadata) and service (server uptime, reaction time for problems, data security, e.g. redundancy, backups).
- **contribute at least 50 permanent operational stations**, with back archive, not previously distributed by other EIDA nodes and maintain a permanent data management center.
- meet requirements set by EPOS. At present, this means ensuring all owners of the networks distributed from the node 1) provide a **formal agreement for data distribution** (following EIDA templates), 2) agree to attach an **open data licence** to the data, and 3) have a **DOI for each network code**.

Any prospective nodes that cannot meet all these requirements are encouraged to inform the EMB Chair. Depending on the difficulty encountered, the EIDA Management Board and ETC will help the applicant to meet the criteria or, if no solution is found, inform the ORFEUS Execom.

…

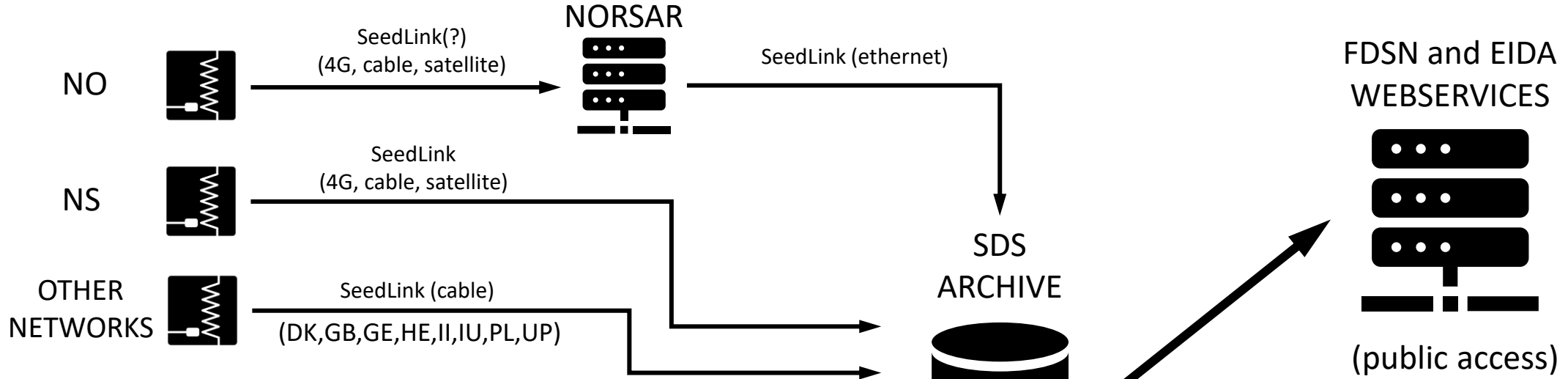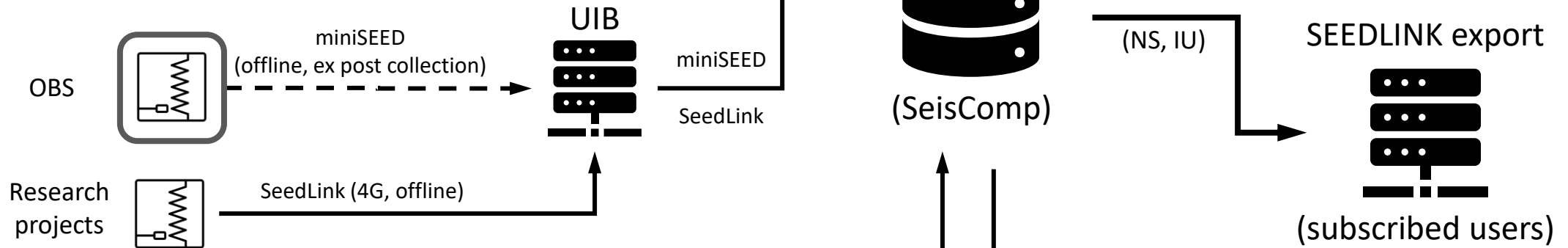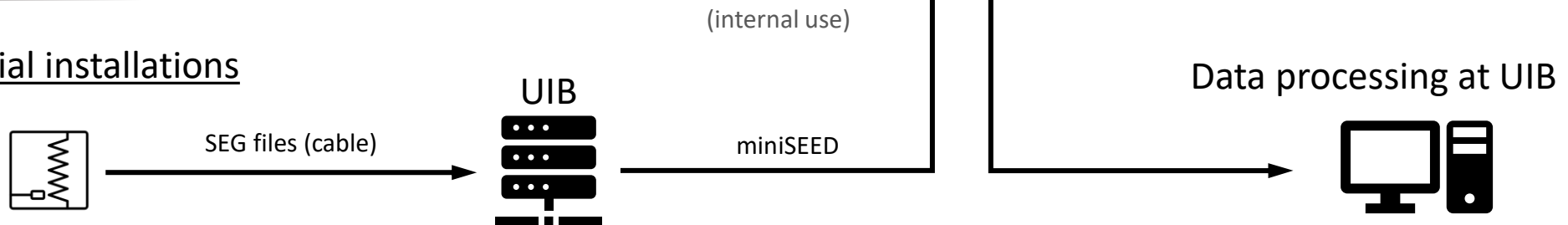# Norwegian EIDA node

UIB-NORSAR

Specific steps done at UIB-NORSAR node

- Preparation of station metadata files
  - Response files are stored as SEISAN or GSE files at UIB → conversion to SEED by gse2seed (Reinoud Sleeman, KNMI) → conversion to inventoryXML → by dlsv2inv (SC) → ingestion to SEISCOMP
- All stations added to SEISCOMP
  - Bindings created
  - Required modules configured (minimal setup)
- Conversion of archive data
  - UIB: Data prior to 2016 needs to be converted to miniSEED (SDS archive)
    - Conversion from SEISAN by using WAVETOOL, checked headers and channel names by MSMOD, moved to SDS by DATASELECT
  - NORSAR: CSS data converted to miniSEED for years 1999-2019

UIB-NORSAR Archive

- The archive from the NS network consists of continuous data that for some stations goes back to 2000 and currently (September 2020) contains 6.6 TB of data (2.1 TB 2000-2016; 4.5 TB since 2016)

- The archive from the NO network is complete back to year 1999 for the arrays (NOA 42 stations, ARCES 25 stations, and SPITS 9 stations) and contains 5.6 TB of data (September 2020).

Services at UIB-NORSAR EIDA node

Web service	Link	Output format / interface
DATASELECT	https://eida.geo.uib.no/fdsnws/dataselect/1	mSEED
STATION	https://eida.geo.uib.no/fdsnws/station/1	StationXML, TXT
WFCATALOG	https://eida.geo.uib.no/eidaws/wfcatalog/1	JSON
WFCATALOG availability	https://eida.geo.uib.no/wfcatalog/availability/	user interface
WFCATALOG metrics	https://eida.geo.uib.no/wfcatalog/metrics/	user interface
WebDC3 Portal	https://eida.geo.uib.no/webdc3/	user interface

Station metadata handling at UIB

- Check SEI files - is content correct?
- Convert all SEI to **GSE (source)**
  - Check epoch definition (time range)
  - Routine **resp** for making new GSE files.
    - Manual addition: sample rate, comment: import from SEI
- Convert GSE to SEED
  - **gseresp2seed** (part of SEISAN) – needs STATION0.HYP (coordinates)
  - concatenating all SEED files into one SEED (works but not sure if this is the right way)
  - **verseed** - validating SEED (https://seiscode.iris.washington.edu/projects/verseed)
- **Stationxml2seed:** https://github.com/iris-edu/stationxml-seed-converter (JAVA)
  - **stationXML validator** from IRIS https://github.com/iris-edu/StationXML-Validator/wiki (JAVA)
- Importing to SC (**dlsv2inv**: SEED to inventoryXML)
- Might be interesting to compare FDSN STATION web service output to original stationXML for verification

- Yasmine stationXML editor: https://github.com/iris-edu-int/yasmine-stationxml-editor/graphs/contributors

Discussion

- How do you manage your station metadata currently?
- Are you technically ready to become an EIDA node?