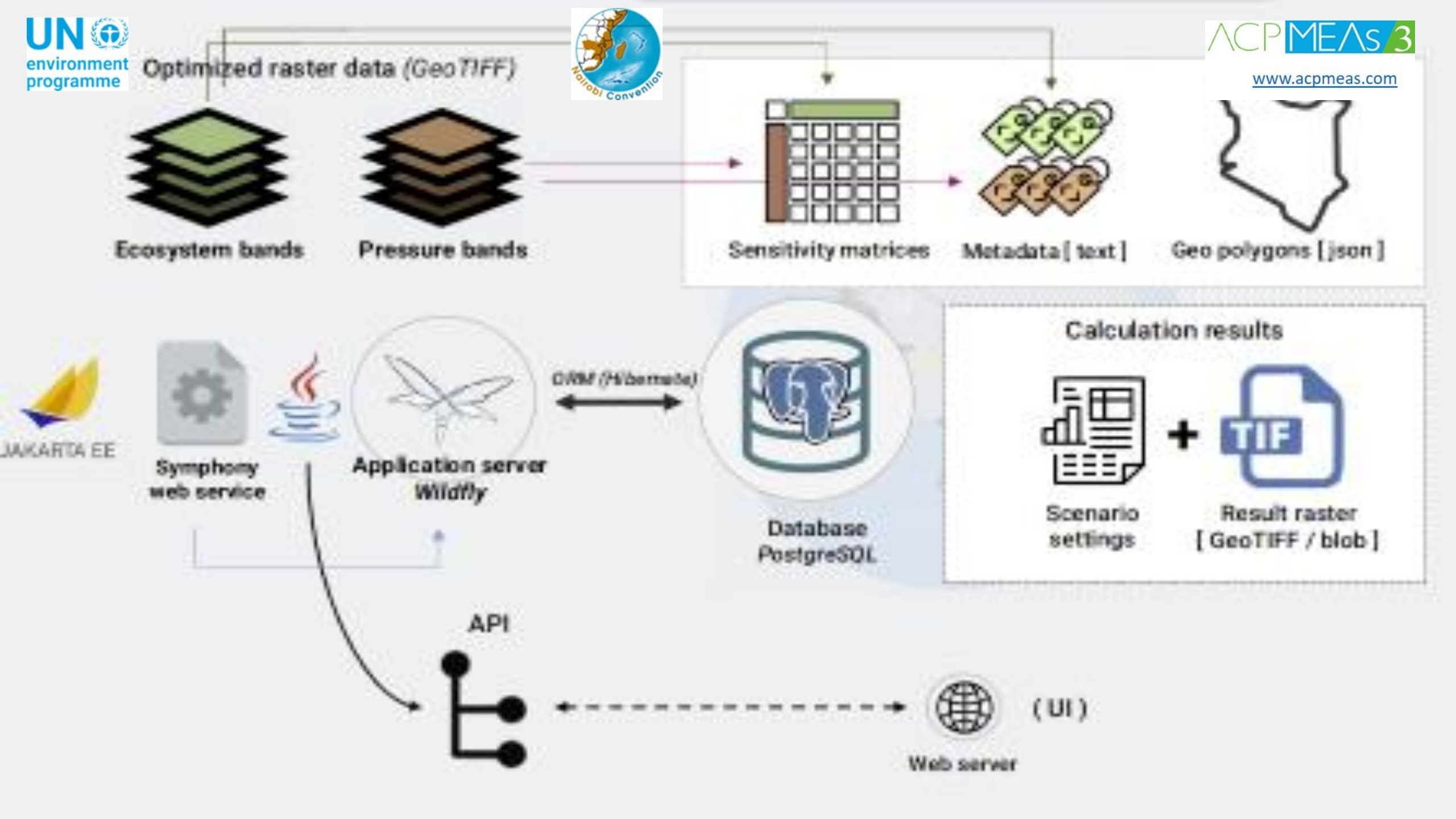
WESTERN INDIAN OCEAN SYMPHONY TOOL

DEPLOYMENT INFRASTRUCTURE

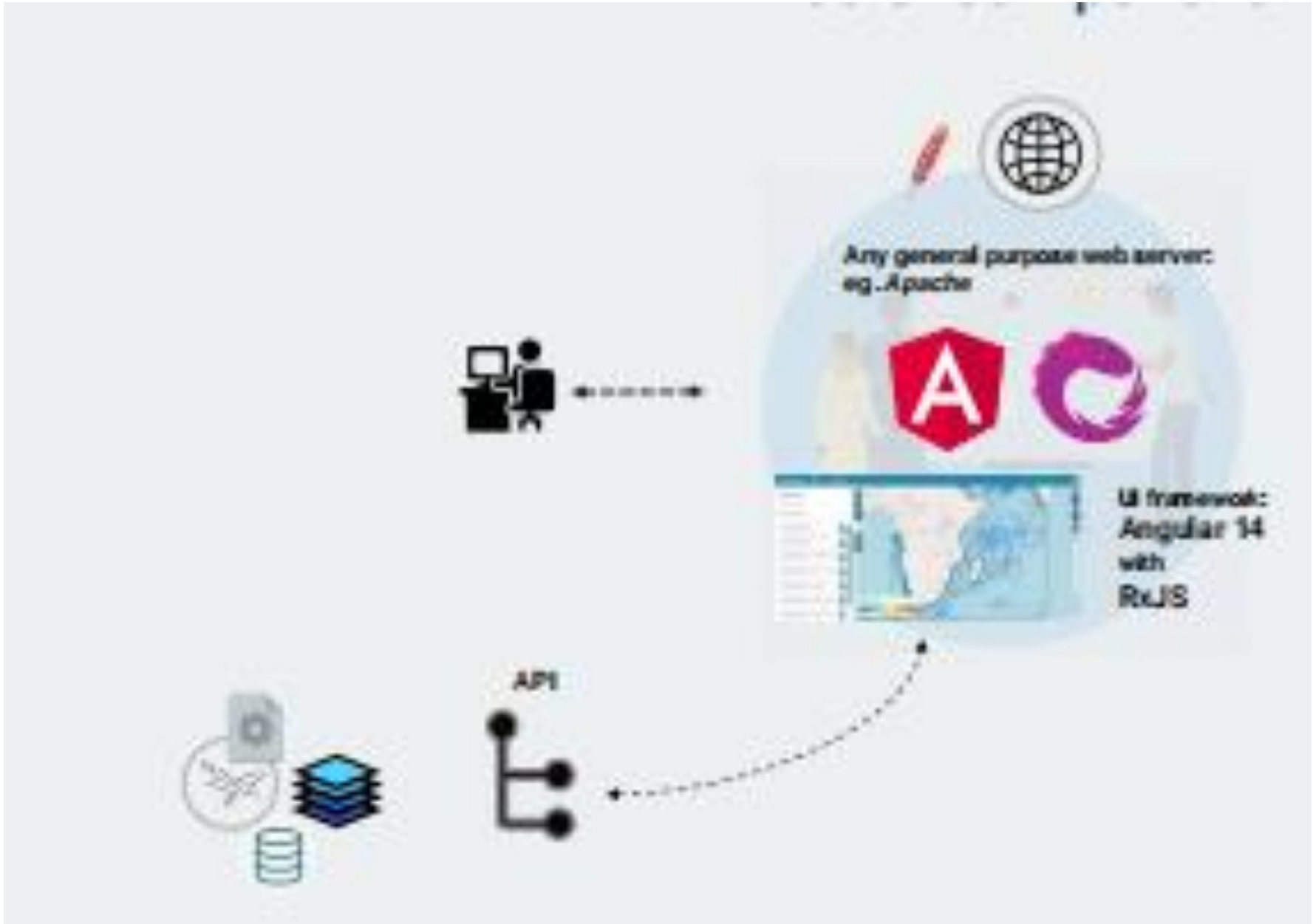
WIO Symphony Architecture

- Jakarta EE – based app
- Wildfly 26 app server
  - Configurable identity management
- Hibernate ORM
- PostgreSQL 14
- WIO specific:
  - Database growth is entirely due to result rasters being saved as binary blobs, especially large areas


Optimized raster data (GeoTIFF)
Ecosystem bands
Pressure bands
Sensitivity matrices
Metadata [text]
Geo polygons [json]
Calculation results
Scenario settings
Result raster [GeoTIFF / blob]
JAKARTA EE
Symphony web service
Application server Wildfly
ORM (Hibernate)
Database PostgreSQL
API
(UI)
Web server

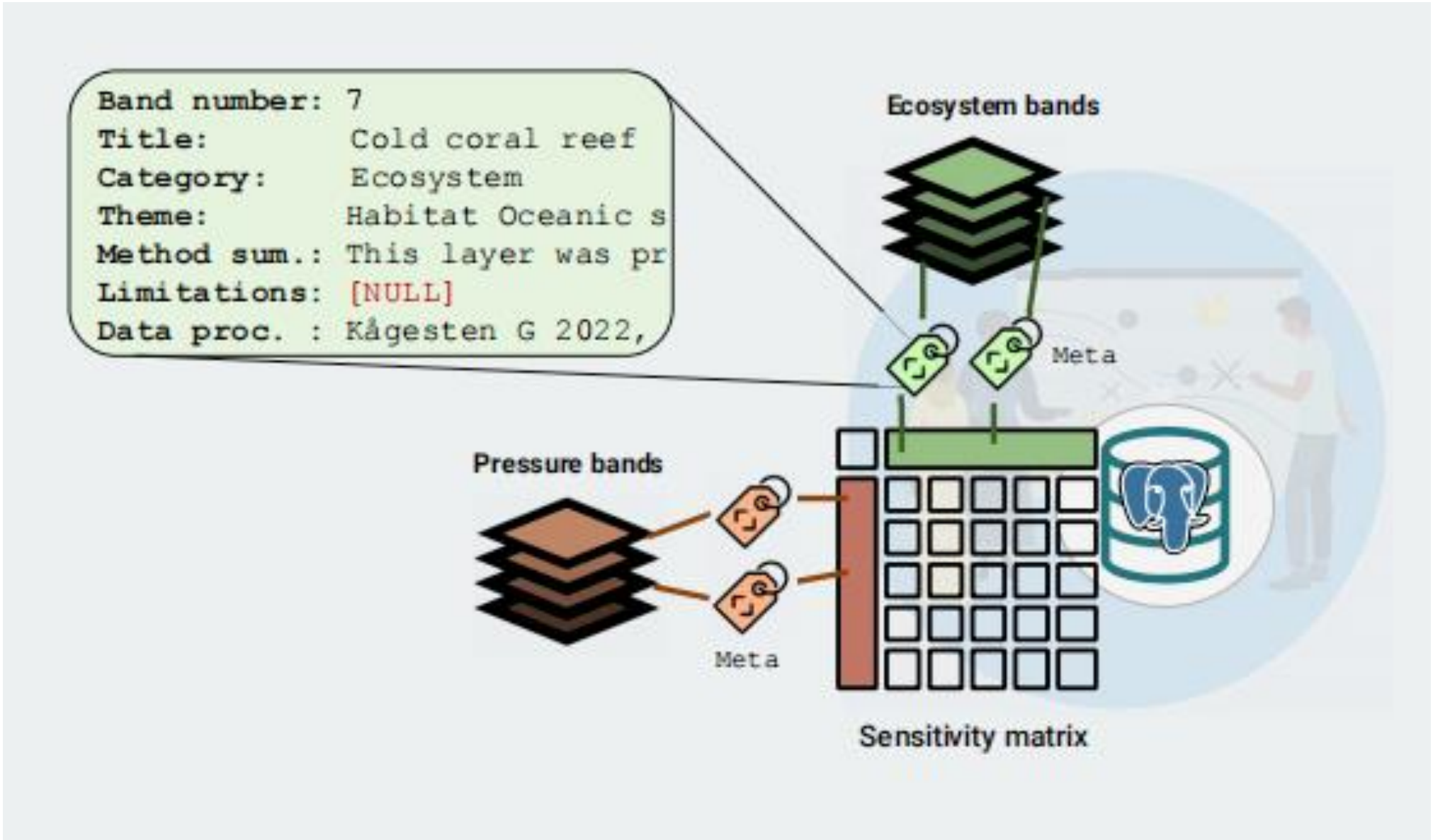
WIO Symphony: Web UI component

- Web UI built with Angular 14
- Utilizes RxJS for state management
- Utilizes translation module:
  - ngx-translate
  - Currently maintained translations – English, French, Swedish
- WIO Symphony only diverges from the main branch in the UI, owing to special requirements for a number of graphical details: attribution clauses on calculation report sheet and ”About” dialog box + the NC logotype


Any general purpose web server:
eg. Apache
UI framework:
Angular 14
with
RxJS
API

WIO Symphony: Raster data↔Database

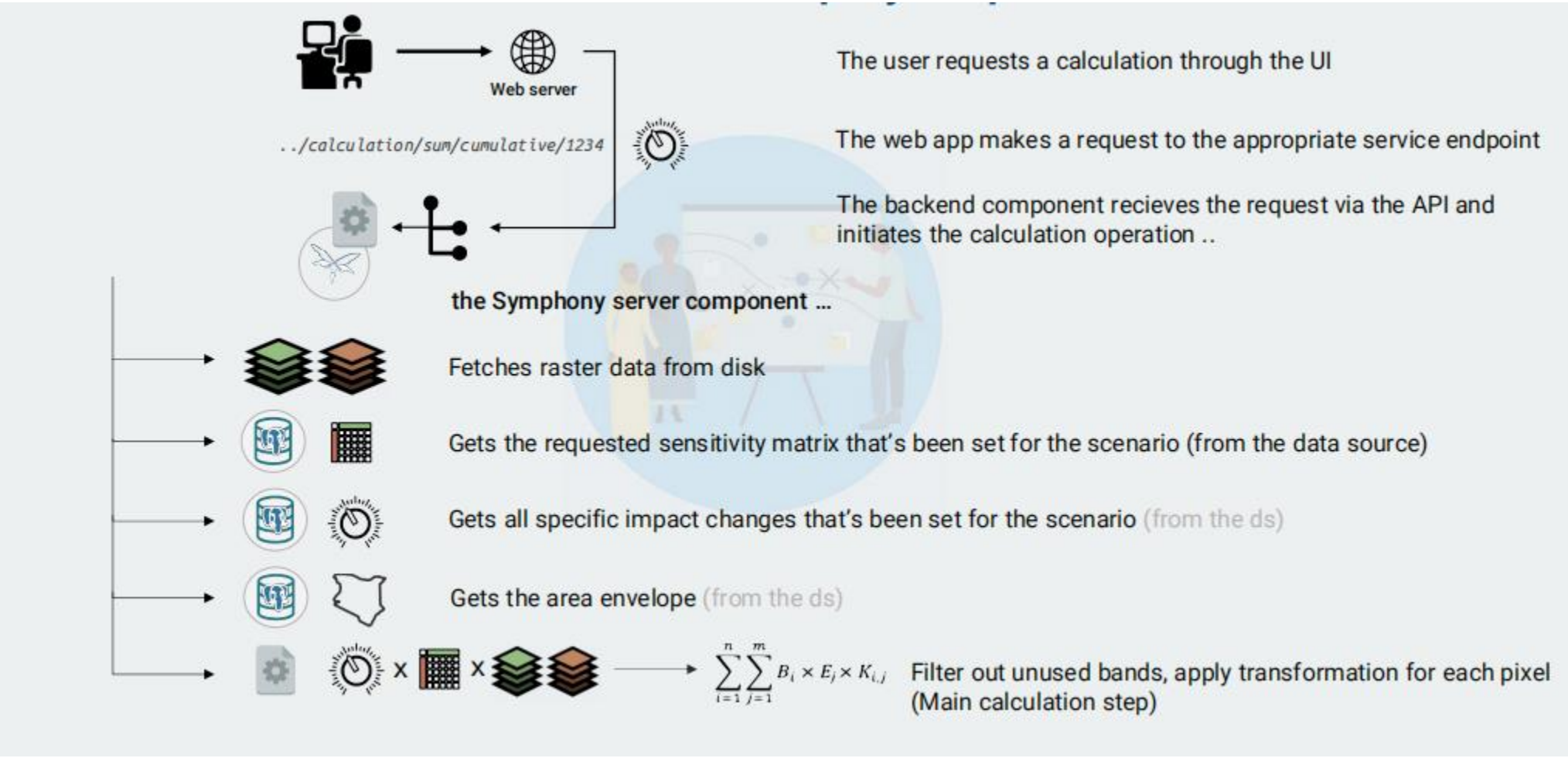
- Raster bands are intertwined with matrix settings
- Note that there may be multiple read_x0002_only matrices defined for a baseline (but always one default matrix)
- Metadata corresponds to specific baseline dataset and provides keys for matrices
- Currently there are nine meta fields in active use by Symphony:
  - Band number
  - Title (eg. ”Cold coral reef”)
  - Category (Pressure / Ecosystem)
  - Theme (eg. ”Habitat Oceanic seafloor”)
  - Method summary
  - Limitations
  - Value range
  - Data processing details
  - Data sources (list)


Band number: 7
Title: Cold coral reef
Category: Ecosystem
Theme: Habitat Oceanic s
Method sum.: This layer was pr
Limitations: [NULL]
Data proc. : Kågesten G 2022,
Ecosystem bands
Meta
Pressure bands
Meta
Sensitivity matrix

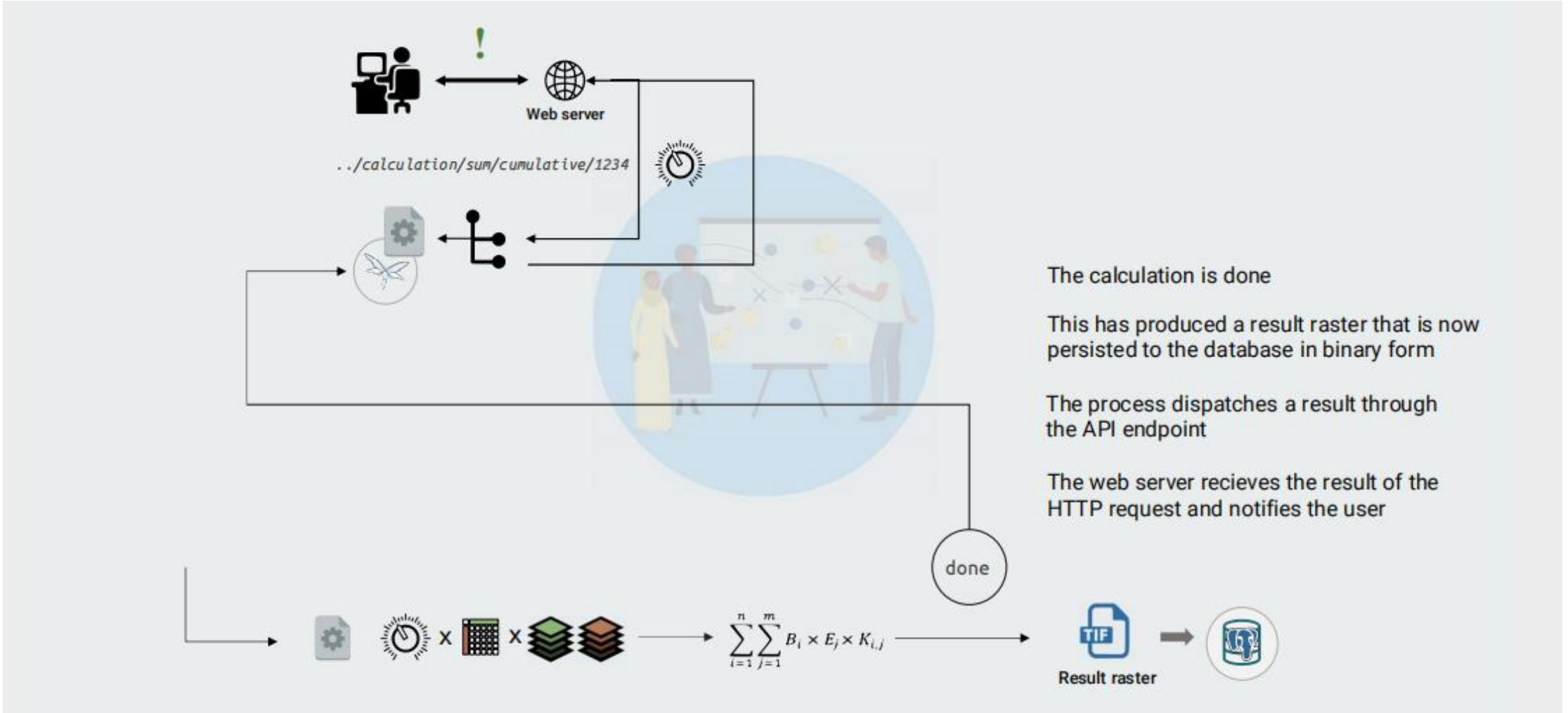
WIO Symphony: polygons / boundaries

- Symphony uses three different types of polygons:
  1. Read-only area polygons
  2. User-defined polygons (created by ”free hand” drawing on map, or by GeoPackage import)
  3. Calculation area boundaries
- Read-only areas are listed on the top section under ”Areas” tab, available to all users. • User-defined areas are listed below the Read-only areas and are available per user. • Boundaries are used by the application to determine system behaviours.
- Most prominent is the association between a Calculation Area and its default sensitivity matrix.
- Symphony can accomodate different Calculation Areas, to differentiate specific matrices for multiple areas. SwAM instance for example defines three Calculation Areas. However, this feature isn’t used in WIO Symphony – the entire grid shares a common default sensitivity matrix, that is, there is only one Calculation Area defined, aptly called ”Whole grid”.
- All selectable areas must be contained in a Calculation Area

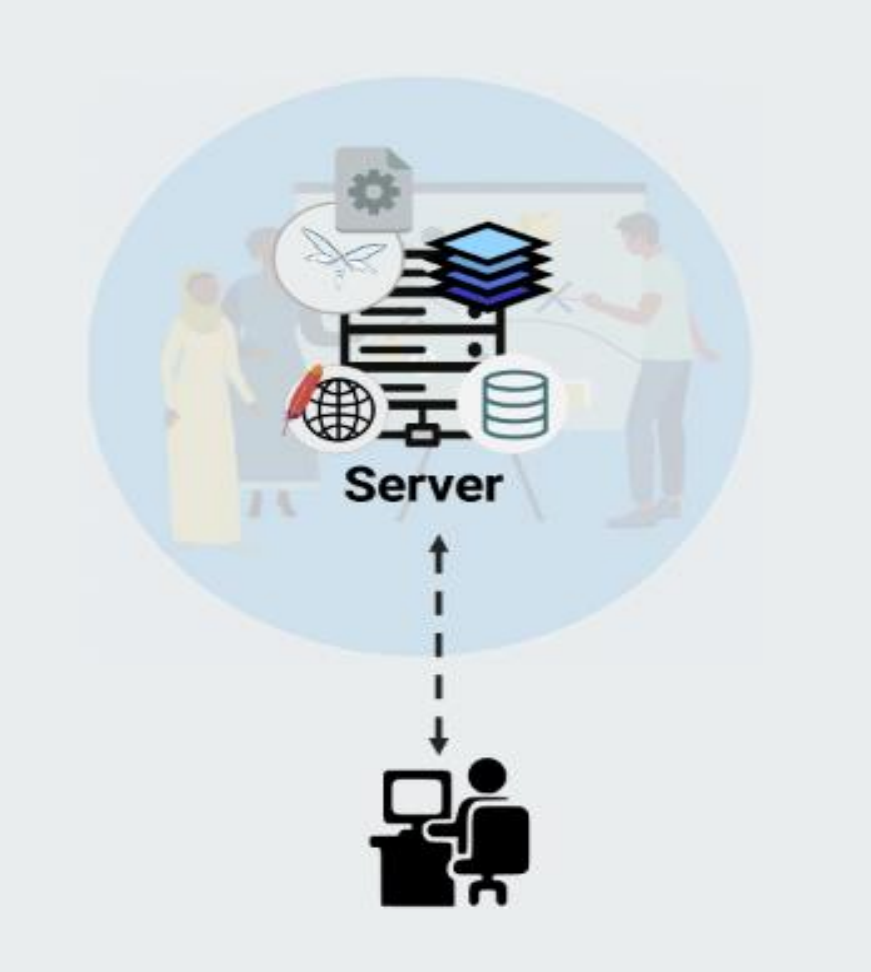
WIO Symphony: Architectural overview
Calculation step by step



WIO Symphony: Architectural overview
Calculation step by step



WIO Symphony Current Setup



- ”Crowded” setup
- Inappropriate for a live production environment
- Workload for each component (calculation/traffic/data retrieval) affects entire system
- Problematic for maintenance

WIO Symphony Future Setup

- Minimal setup: combines application and web server
- Database server on the same Cloud Space (UN Cloud - AZURE)
- Optimally: usage of ”own” server infrastructure (Scalable so when the load is heavy it accomodates the requests)

Asante