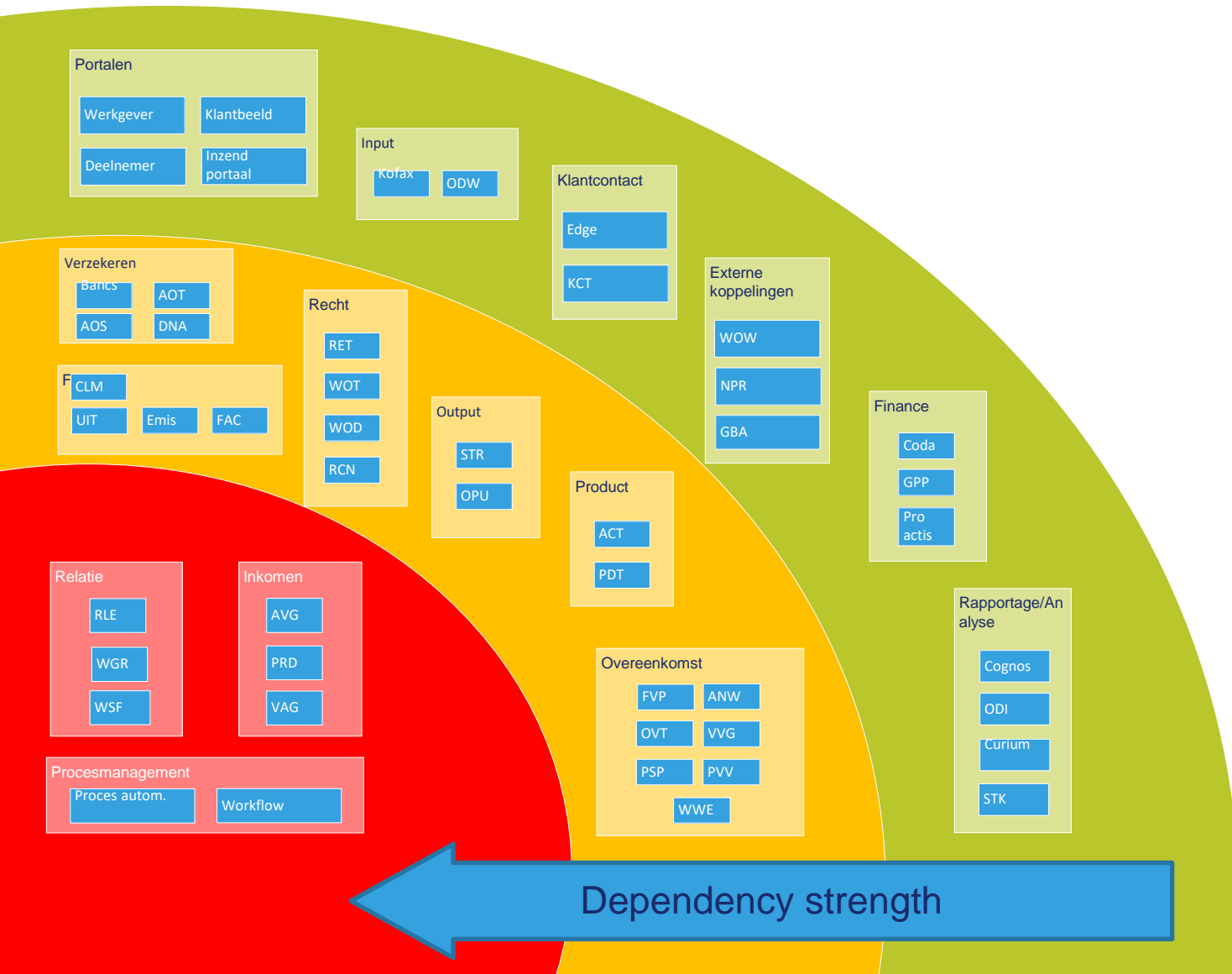


SpronQ



Transforming a large legacy monolith

Application landscape



The Database !

The screenshot displays a database management interface. On the left, a sidebar shows a tree view of tables under the 'Tables' folder. The table names are listed as follows:

- COMP_RLE.RLE_AANMELDINGEN
- COMP_RLE.RLE_ADMINISTRATIES
- COMP_RLE.RLE_ADRESSEN
- COMP_RLE.RLE_ADRESSEN_JN
- COMP_RLE.RLE_AFSTEMMINGEN_GBA
- COMP_RLE.RLE_BESTANDEN
- COMP_RLE.RLE_BESTANDSABONNEMENTEN
- COMP_RLE.RLE_BESTAND_CASES
- COMP_RLE.RLE_BRIEVEN_KVK
- COMP_RLE.RLE_BRON_EX_PARTNERS
- COMP_RLE.RLE_BULK_BERICHTEN_EXT
- COMP_RLE.RLE_CAMPAGNES
- COMP_RLE.RLE_CAMPAGNE_DEFINITIES
- COMP_RLE.RLE_COMMUNICATIE_CATEGORIEEN
- COMP_RLE.RLE_COMMUNICATIE_CAT_KANALEN
- COMP_RLE.RLE_COMMUNICATIE_NUMMERS
- COMP_RLE.RLE_CONTACTPERSONEN
- COMP_RLE.RLE_CONTACTPERSONEN_JN
- COMP_RLE.RLE_DUBBELE_FAKTOREN
- COMP_RLE.RLE_EXP_HUWELIJKEN_GBAV
- COMP_RLE.RLE_EXP_XML_FOUTEN_GBAV
- COMP_RLE.RLE_EXT_ADRESSEN_BESTAND
- COMP_RLE.RLE_EXT_ADR_SUBSETS
- COMP_RLE.RLE_EXT_ADR_VERWERKINGSREGELS
- COMP_RLE.RLE_EXT_AFGESTEMDE_ADRESSEN
- COMP_RLE.RLE_EXT_KANAALVOORKEUREN
- COMP_RLE.RLE_FINANCIEL_NUMMERS
- COMP_RLE.RLE_FINANCIEL_NUMMERS_JN
- COMP_RLE.RLE_FINNR_CODA
- COMP_RLE.RLE_GBAV_EEN
- COMP_RLE.RLE_GBAV_FOUT
- COMP_RLE.RLE_GBAV_MEER
- COMP_RLE.RLE_GBA_HUWELIJKEN_TNT
- COMP_RLE.RLE_GBA_INDIC_EXT
- COMP_RLE.RLE_GBA_INDIC_TMP
- COMP_RLE.RLE_GBA_IST_EXT

The main area of the interface shows a complex diagram of table relationships. Red lines connect various tables, indicating foreign key relationships. Several tables are highlighted with red boxes, including 'COMP_RLE.RLE_AANMELDINGEN', 'COMP_RLE.RLE_ADMINISTRATIES', and 'COMP_RLE.RLE_ADRESSEN'. The diagram illustrates a dense network of relationships between numerous tables, with some tables appearing as multiple overlapping instances.

Usual 2000s architecture

Web

Forms

Interface

WS WS WS WS WS WS WS

Service Bus

Logic

Stored procedures

RL AV PV RE
E G V T

Database



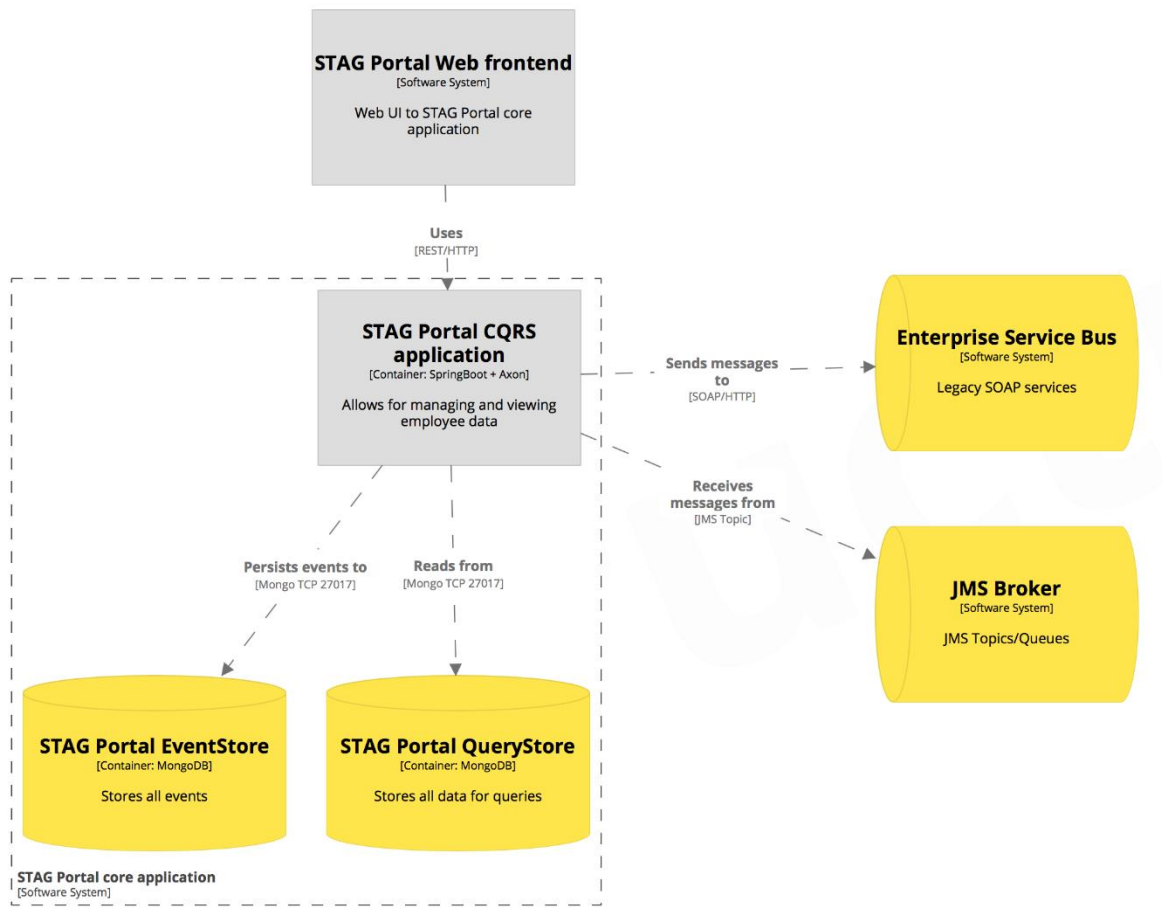
Monolith

- Can only be tested as a whole
- Changes on one end can break functionality on another end
- Can only be deployed as a whole (limits throughput)

Now What ?

- Build a new web application for communication with employers
- Limits rework on the current monolith (too complex)
- Allows renewal based on clean data
- Keeps the old stuff running

C4 Context



Integration

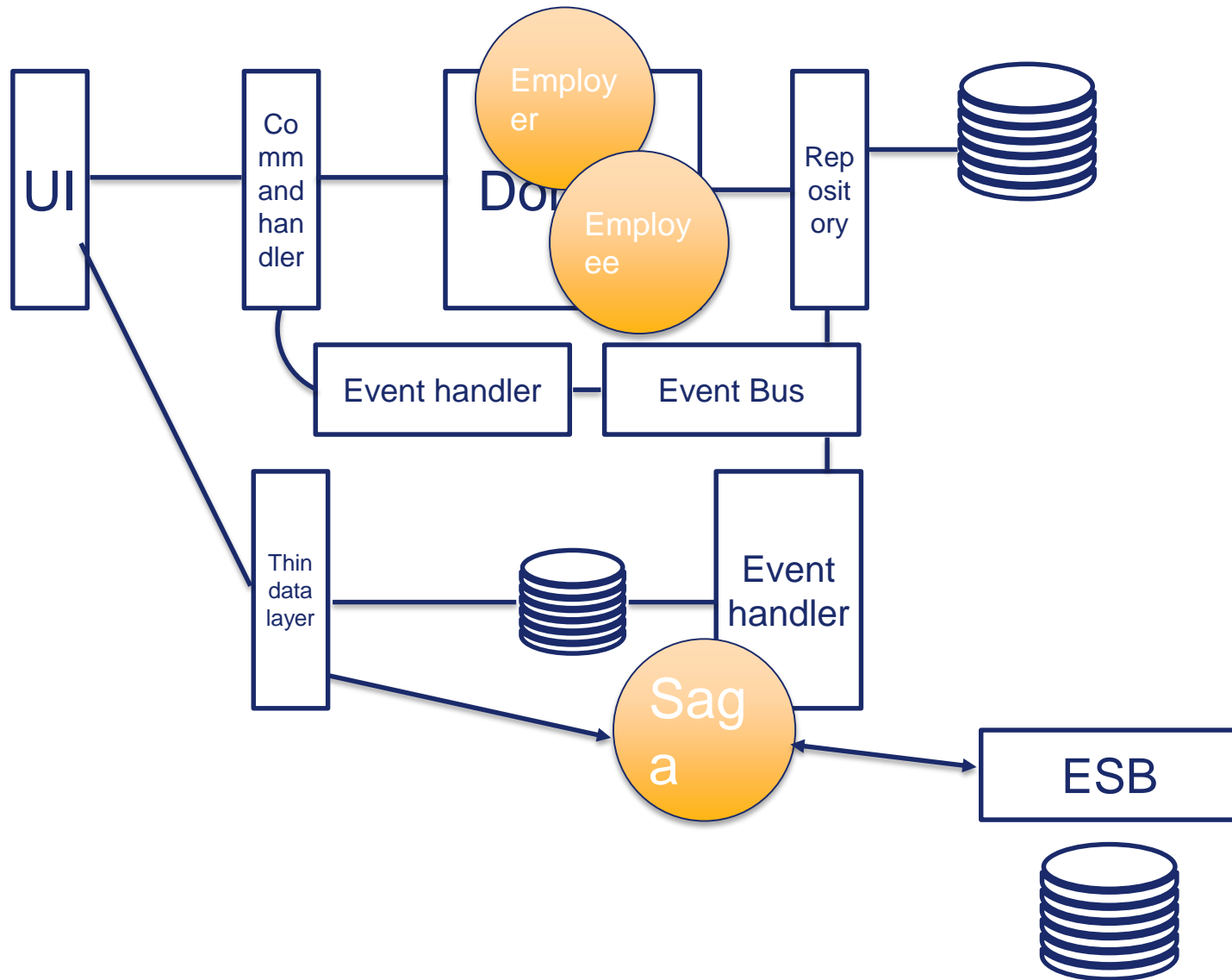
- Logic executes and changes given data
- Employers must see what they have entered
- Employers are not allowed to see certain changes (privacy)
- Acceptance of changes may take up to 5 days

Anti Corruption Layer

- Integration of long running transactions (days not seconds)
- Changes may appear due to outside input (multi channel)
- Axon Saga to keep track of mutations

- “No migration” approach
- Clear ‘business’ capability to handle unknowns

'Axon standard picture'



The other goal

- Improve the performance of software delivery
- Show that a microservices architecture has positive effect
 - On Lead Time
 - On Deployment Frequency
 - On Team Autonomy

Shift Left

- Next to Unit tests
- Coverage %
- Architecture diagram evolution (C4)
- Architecture structure protection
- OWASP execution
- Load test execution
- Automated approval of compliance processes

SpronQ



Questions ?

olger.warnier@spronq.com