UBIQUITOUS LANGUAGE

FROM BEGINNER TO EXPERT IN 30 MINUTES

BJÖRN GENFORS & BJÖRN BESKOW

CADEC 2022.02.02 | CALLISTAENTERPRISE.SE

CALLISTA

BJÖRN & BJÖRN

BJÖRN & BJÖRN

AGENDA

- Introduction
  - Ubiquitous language - what and why?
- Part 1: Theory
  - Today's terminology
  - The terminology of terminology
  - Ontologies: what, why & how
- Part 2: Ubiquitous language in Domain Driven Design (DDD)
  - Establishing a common language
  - Refining the language into a Domain Model
  - Bounded Contexts and Context Maps

UBIQUITOUS LANGUAGE

UBIQUITOUS LANGUAGE - WHAT AND WHY

- What is it?
  - Part of Domain Driven Design (DDD)
  - Pervasive language

- Why?
  - Language is the basics for common understanding
  - Nothing lost in translation

Adapted from source: https://upload.wikimedia.org/wikipedia/commons/2/23/Rosetta_Stone.JPG

TODAY'S TERMINOLOGY

TODAY'S TERMINOLOGY

- The content
- The container
- The art of putting content in the container


Lexicon
Taxonomy
Terminology
Code system
Controlled vocabulary
Dictionary
Classification
Thesaurus
Enumerated type
Nomenclature
Glossary
Ontology

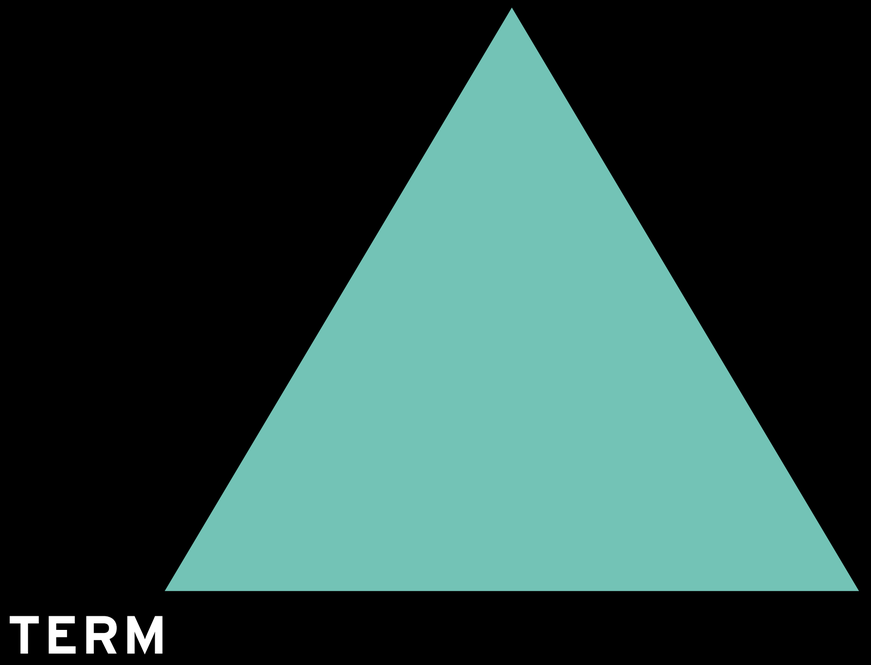
TODAY'S TERMINOLOGY

- The content: controlled vocabulary
- The container (+ content): ontology
- The art of putting content in the container: terminology

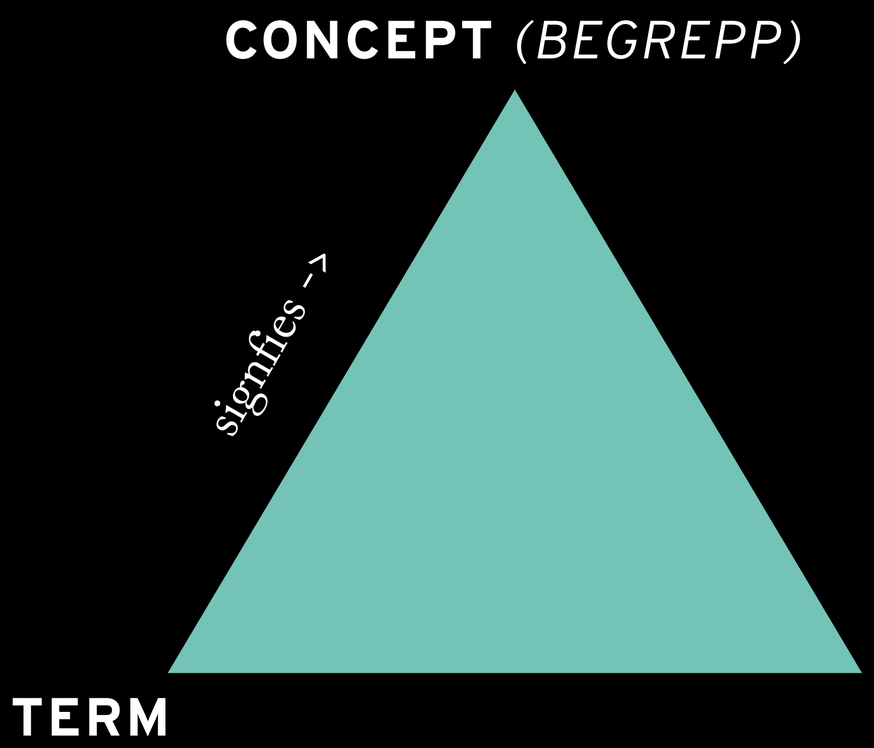
THE TERMINOLOGY OF TERMINOLOGY

CALLISTA

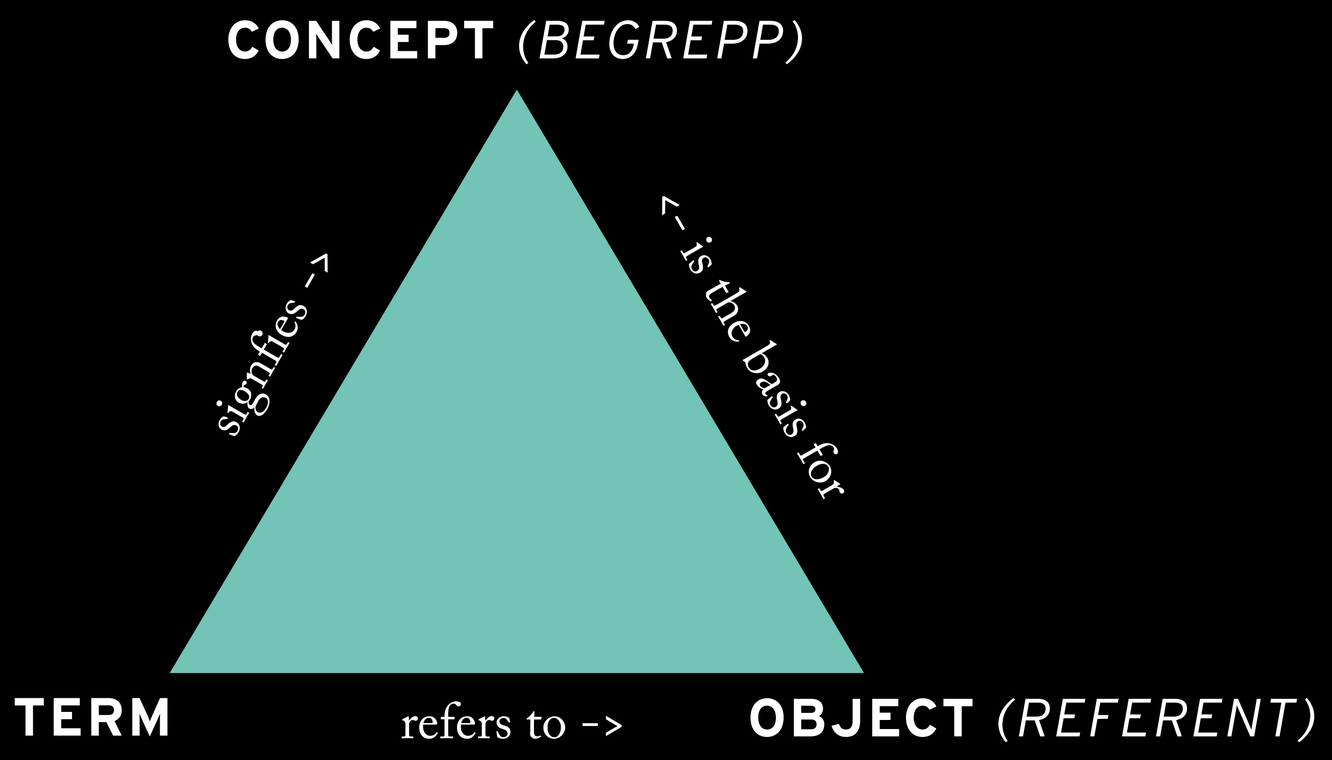
THE TERMINOLOGY OF TERMINOLOGY - THE SEMANTIC TRIANGLE



THE TERMINOLOGY OF TERMINOLOGY - THE SEMANTIC TRIANGLE



THE TERMINOLOGY OF TERMINOLOGY - THE SEMANTIC TRIANGLE



THE SEMANTIC TRIANGLE - AN EXAMPLE

CONCEPT

signifies ->

<- is the basis for

TERM

refers to ->

OBJECT

"APPLE"

THE ART OF TERMINOLOGY - HOW

1. Focus on the concept. ”What is this thing we’re talking about”
2. Figure out a good term. ”What do we call this thing we’re talking about”

What’s in a name? That which we call a rose would by any other name smell as sweet.
- Juliet / William Shakespeare

THE ART OF TERMINOLOGY - WHY

- Resolves language issues with
  - Synonymy
    - » intelligent/smart
  - Homonymy
    - » Bar
    - » Ring
  - Polysemy (non-accidental homonymy)
    - » Stomach

- (The opposite of NLP)

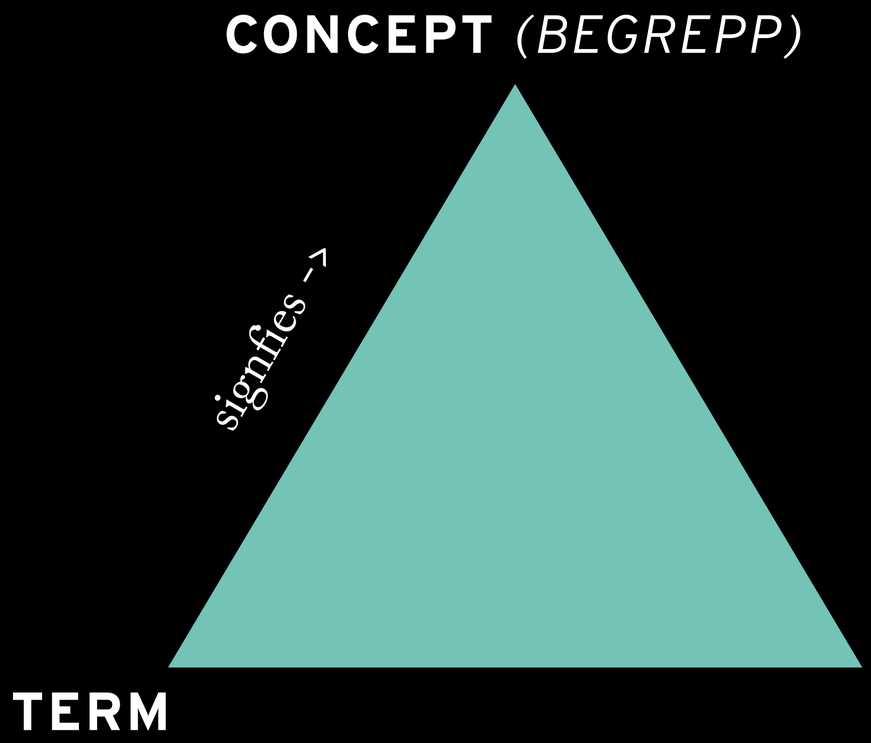
DESIRED PROPERTIES OF ONTOLOGIES

ONTOLOGIES, WHAT AND WHY?

- In philosophy of science: a representation of reality
- Formalism describing knowledge
  - Enables machine processable semantics
- Examples:
  - SNOMED CT
  - The semantic web

CONCEPT ORIENTATION

- What is concept orientation?
  - Separate concepts from terms
  - Focus on the concept, not the term



CONCEPT ORIENTATION, CONTINUED

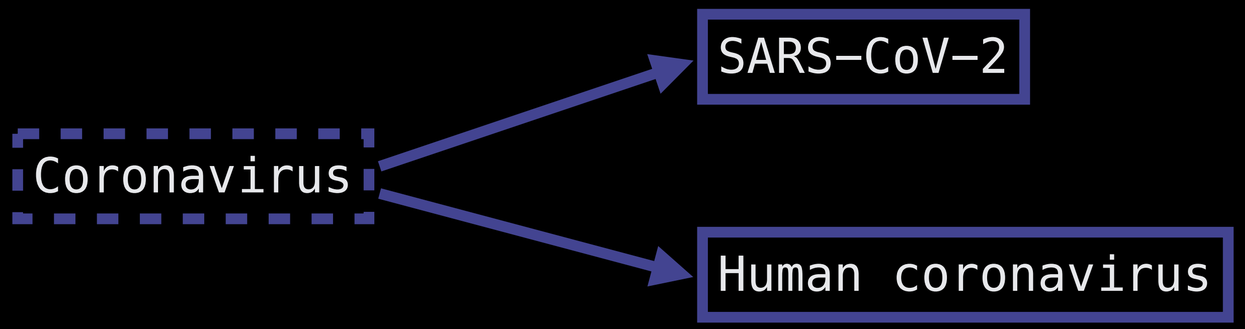
- Why should you do it?
  - Goes back to the basics of terminology
    - » Synonymy
    - » Homonymy
    - » Polysemy

CONCEPT ORIENTATION, CONTINUED

- Why should you do it?
  - Goes back to the basics of terminology
    - » Synonymy
    - » Homonymy
    - » Polysemy

CONCEPT ORIENTATION, CONTINUED

- Why should you do it?
  - Goes back to the basics of terminology
    - » Synonymy
    - » Homonymy
    - » Polysemy



NON-SEMANTIC IDS

- Examples of semantics in id:s
  - Birthdates
  - Gender identity
  - Serial number
  - Hierarchical position
  - Organizational ownership

Source: transportstyrelsen.se

- Avoidable problems are obvious at second thought

CONCEPT PERMANENCE

- Semantic stability
  - You may **never**:
    - » change the meaning of a concept
  - You may:
    - » Change preferred term
    - » Make concepts redundant

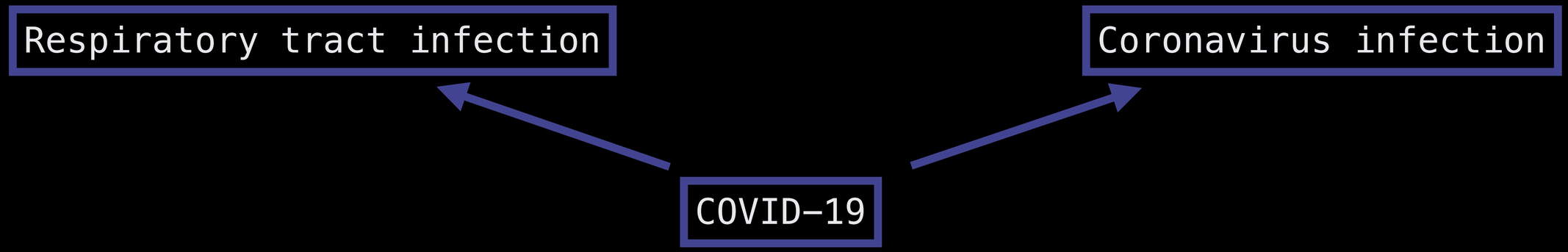
- Avoids semantic drift
  - Ensures historic accuracy/comparability

POLYHIERARCHY

- Every concept may inherit from *one or more* other concepts
  - i.e. can have several *is a*-relationships

- Necessary if concepts are categorized in more than one dimension

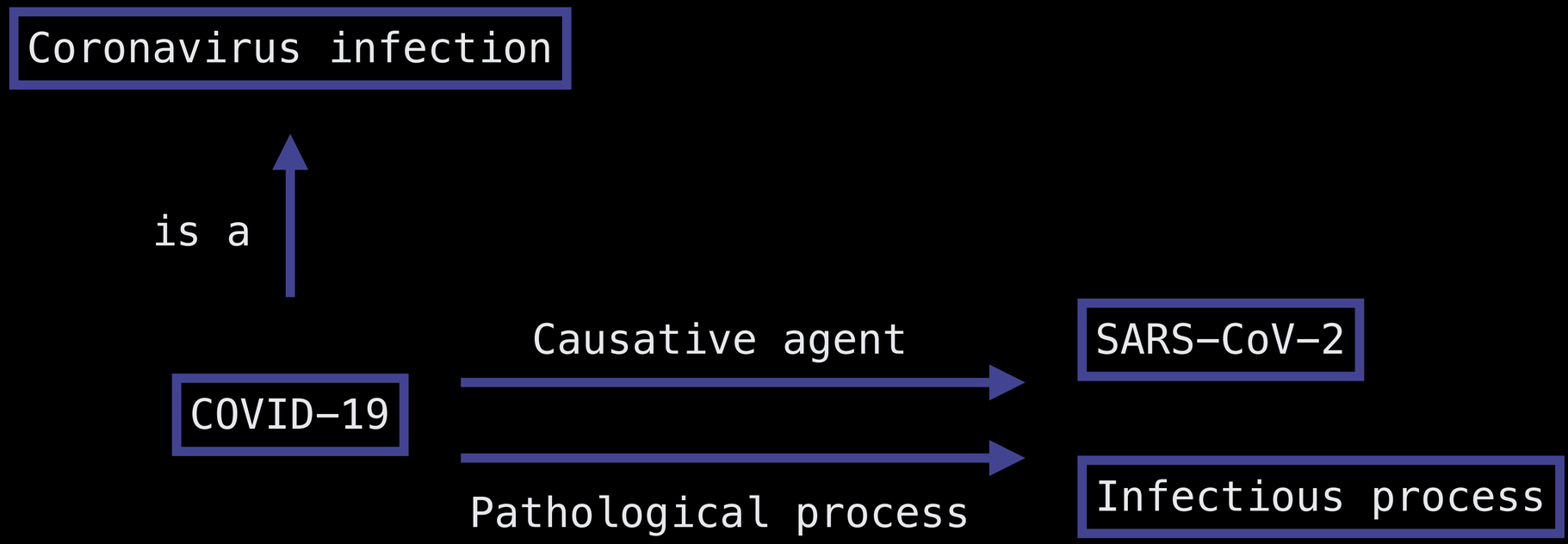
POLYHIERARCHY



FORMAL DEFINITIONS

- Define concepts by their relations to other concepts
- Enables machine processable semantics

FORMAL DEFINITIONS, CONTINUED



THE UBIQUITOUS LANGUAGE …

- Domain == *problem space*
- Model == *solution space*
- Domain Model == *Distilled knowledge about the problem and solution space*

MULTIPLE LANGUAGES?

What if multiple human languages are spoken within the organization/team?

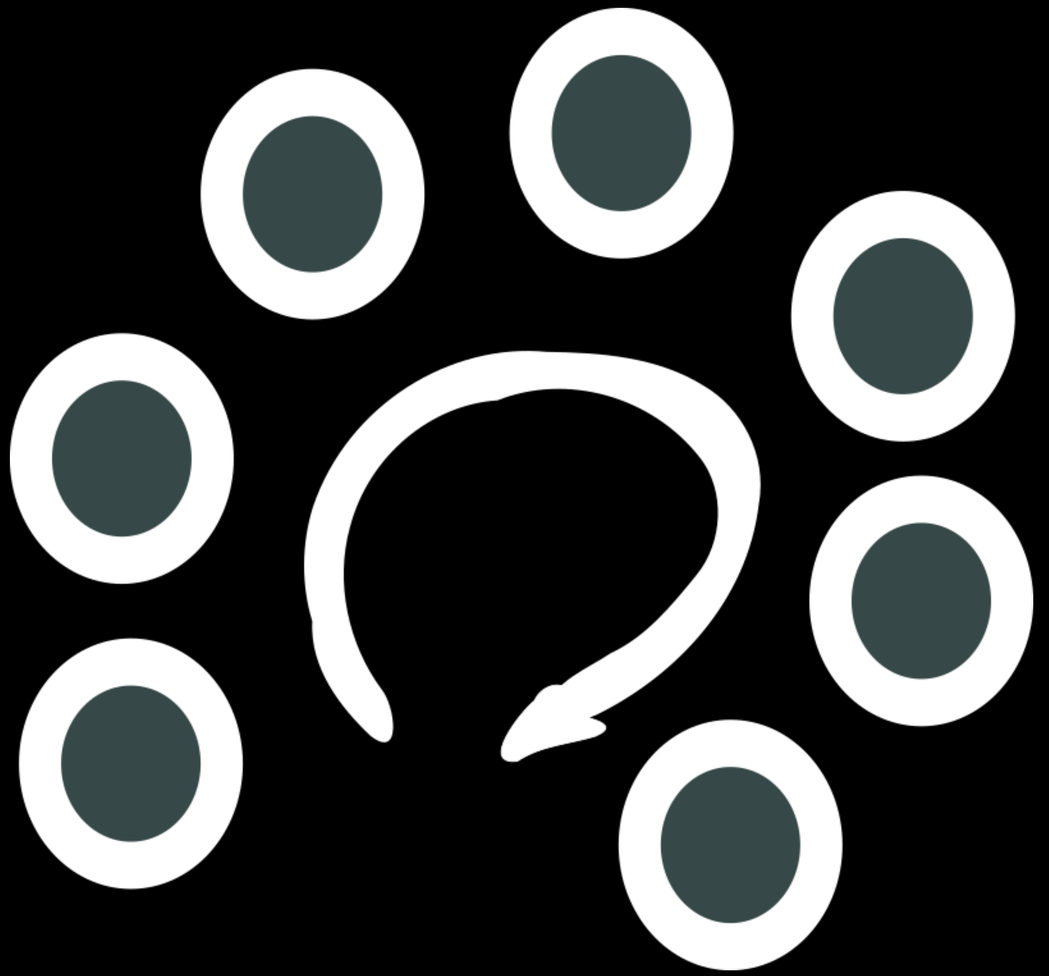
- Decide on one of the human languages as an *Interlingua*, with a formal dictionary to other languages if necessary

A FICTIVE EXAMPLE

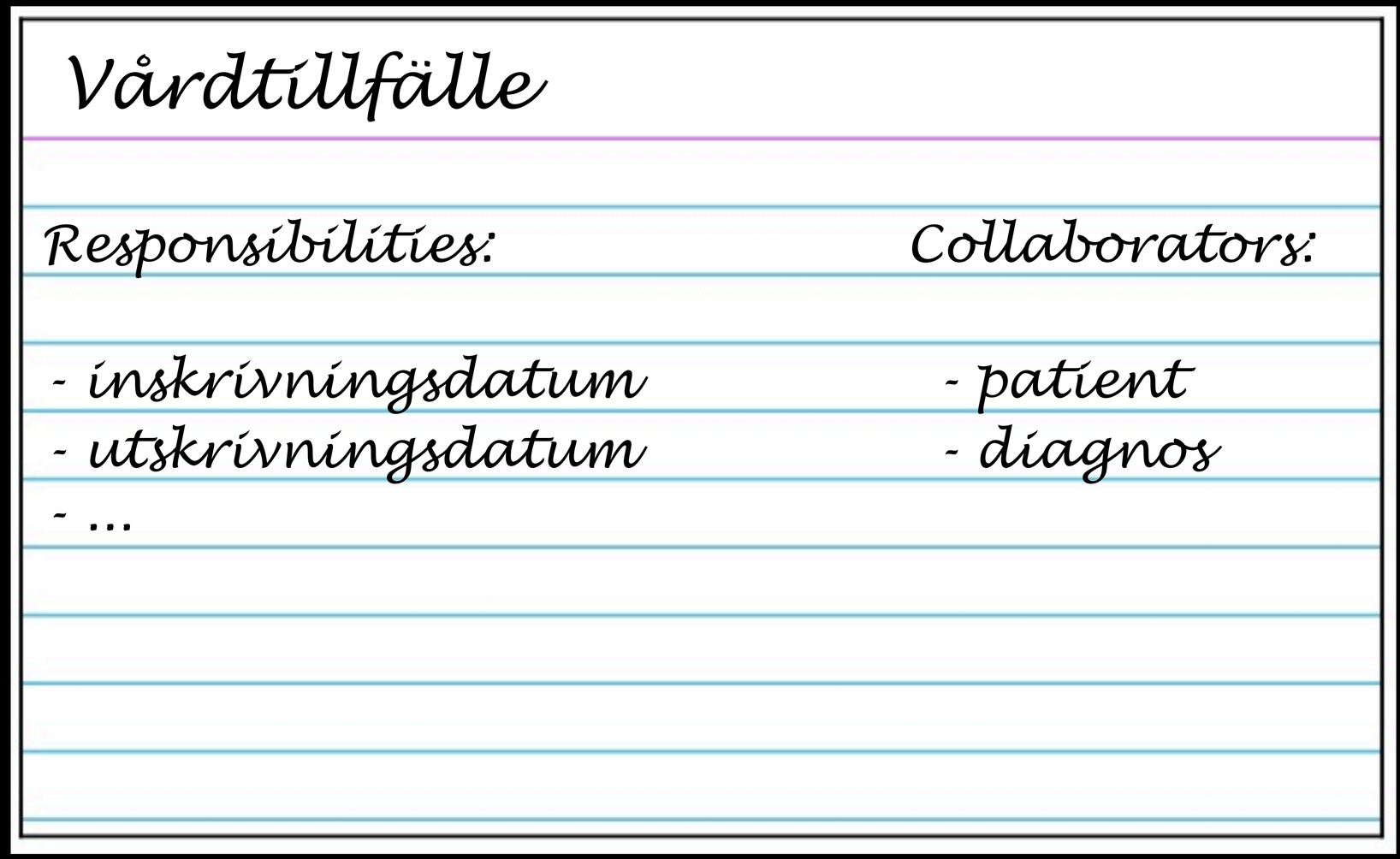
Ett **vårdtillfälle** på **slutenvårdsavdelning** startar när en **patient** efter initial bedömning **skrivs in** på avdelningen med en **primär diagnos**. Vårdtillfället avslutas när patienten **skrivs ut**, t.ex för fortsatt vård inom öppenvården eller i kommunal regi.

REFINING THE LANGUAGE - DISTILLING THE DOMAIN MODEL

- Identify, analyse and define core concepts
  - List their defining characteristics
  - Relate them to other concepts
  - Agree on suitable names
- Organise the concepts into a coherent model
  - Refine the concepts towards a software solution

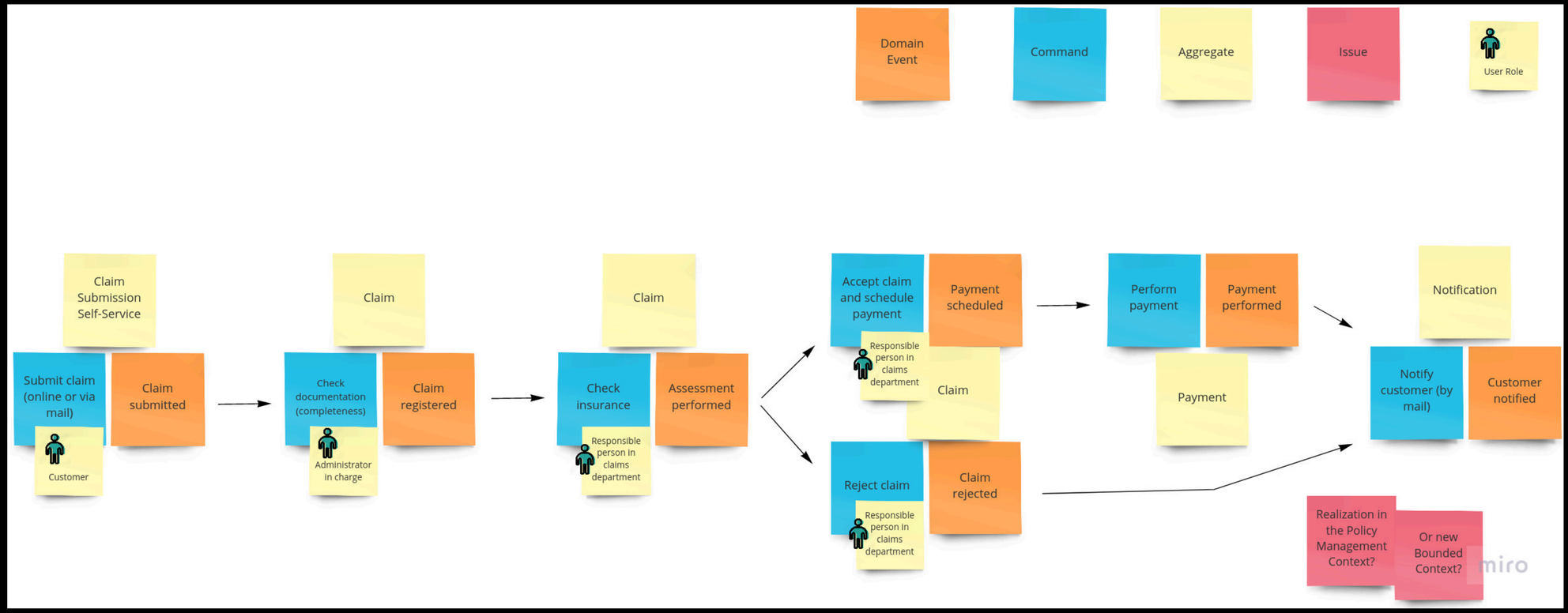


CLASS, RESPONSIBILITY AND COLLABORATORS CARDS (CRC)



https://www.agilealliance.org/glossary/crc-cards/

EVENT STORMING

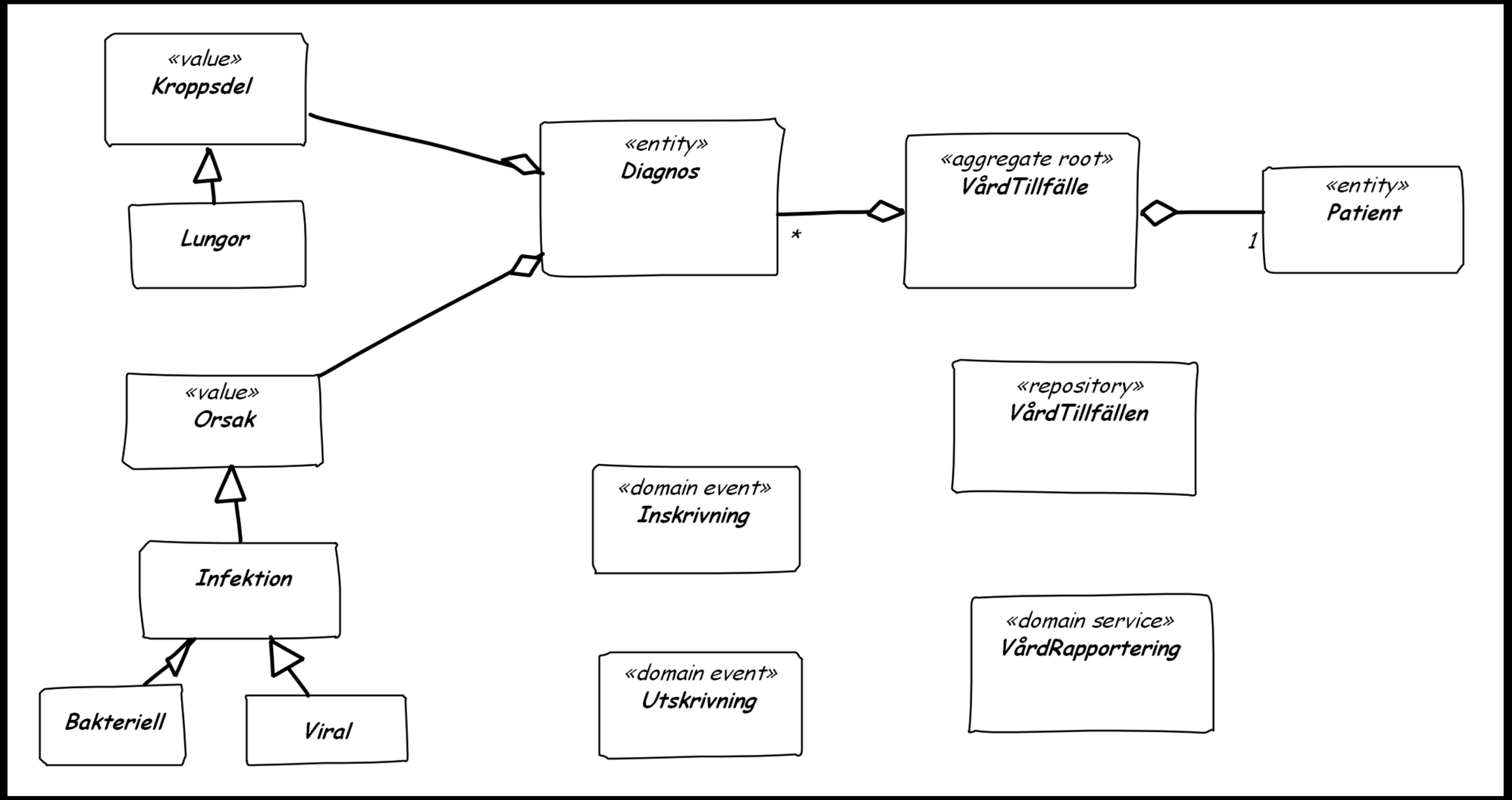


https://leanpub.com/introducing_eventstorming

PROJECT/SYSTEM GLOSSARY

Concept	Definition	Synonyms
Vårdtillfälle	Tidsperiod under vilken en patient ges vård på en sluten avdelning för en primär diagnos och eventuellt ytterligare sekundära diagnoser.	Vårdperiod, Sjukfall
Diagnos	Beskrivning av ett visst sjukdomstillstånd i termer av orsak och fysisk placering	
Patient	…	
Inskrivning	…	

TOWARDS A DOMAIN MODEL



SOFTWARE ARTEFACTS

```
databaseChangeLog:
  - logicalFilePath: db.changelog-1.yml
  - changeSet:
      id: vårdtillfälle
      author: bjobes
      changes:
        - createTable:
            tableName: vård_tillfälle
            columns:
              - column:
                  name: id
                  type: BIGINT
                  constraints:
                    nullable: false
                    primaryKey: true
                    primaryKeyName: vård_tillfälle_pkey
...
```

SOFTWARE ARTEFACTS

```java
@Entity
public class VårdTillfälle {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long vårdTillfälleId;

    @NotNull
    private LocalDate inskrivningsDatum;

    private LocalDate utskrivningDatum;

...

}
```

SOFTWARE ARTEFACTS

```
@Service
public class VårdRapporteringImpl implements VårdRapportering {

  @Inject
  private VårdTillfälleRepository vårdTillfällen;

  @Override
  @Transactional
  public Inskrivning inskrivning(
    Patient patient,
    Diagnos diagnos,
    Avdelning avdelning) {
    ...
 }
 ...
}
```

SOFTWARE ARTEFACTS

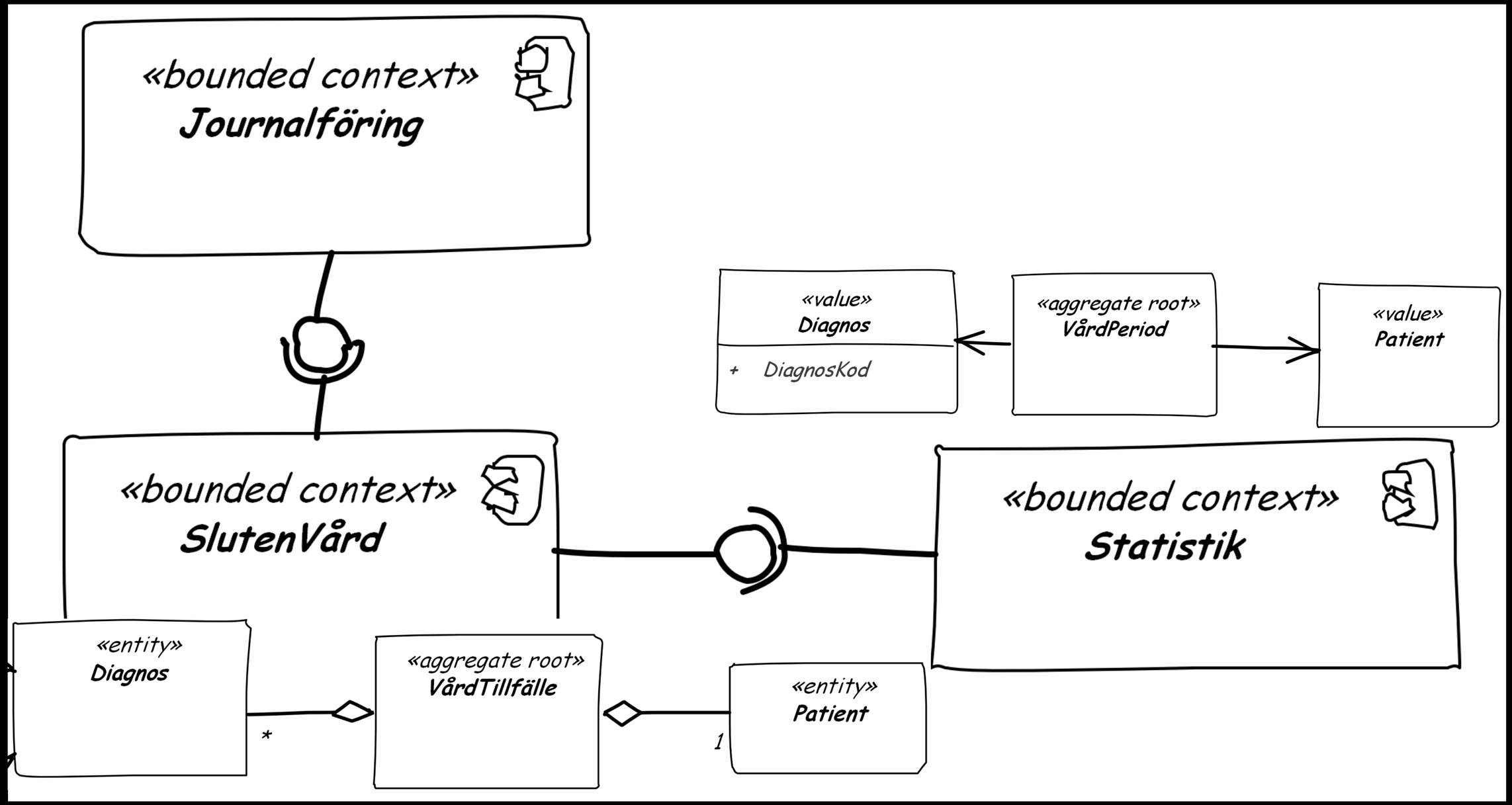
```
public class InskrivningEventConsumer {

  @KafkaListener(topics = “${kafka.topic.event.inskrivning}",
    containerFactory = "listenerContainerFactory")
  public void receive(Inskrivning inskrivning) {
    log.info("Received event {}", inskrivning);
    ...
  }
}
```

SOFTWARE ARTEFACTS

```
{
  "event": "inskrivning",
  "patient": { ... },
  "diagnos": {
    "orsak": { ... },
  },
...
}
```

DEALING WITH CONFLICTS: BOUNDED CONTEXTS, CONTEXT MAPS AND APIS



CONCLUSIONS

- Establishing and refining a Ubiquitous Language is a powerful way to deal with highly complex domains and systems
- The Ubiquitous Language is present in all aspects of and all manifestations of the system
- Efficient reasoning about the domain model is a critical success factor

Time for questions!

BJORN.GENFORS@CALLISTAENTERPRISE.SE
BJORN.BESKOW@CALLISTAENTERPRISE.SE