Towards the Detection of Microservice Patterns Based on Metrics

unibz

João Francisco Lino Daniel

Eduardo Martins Guerra

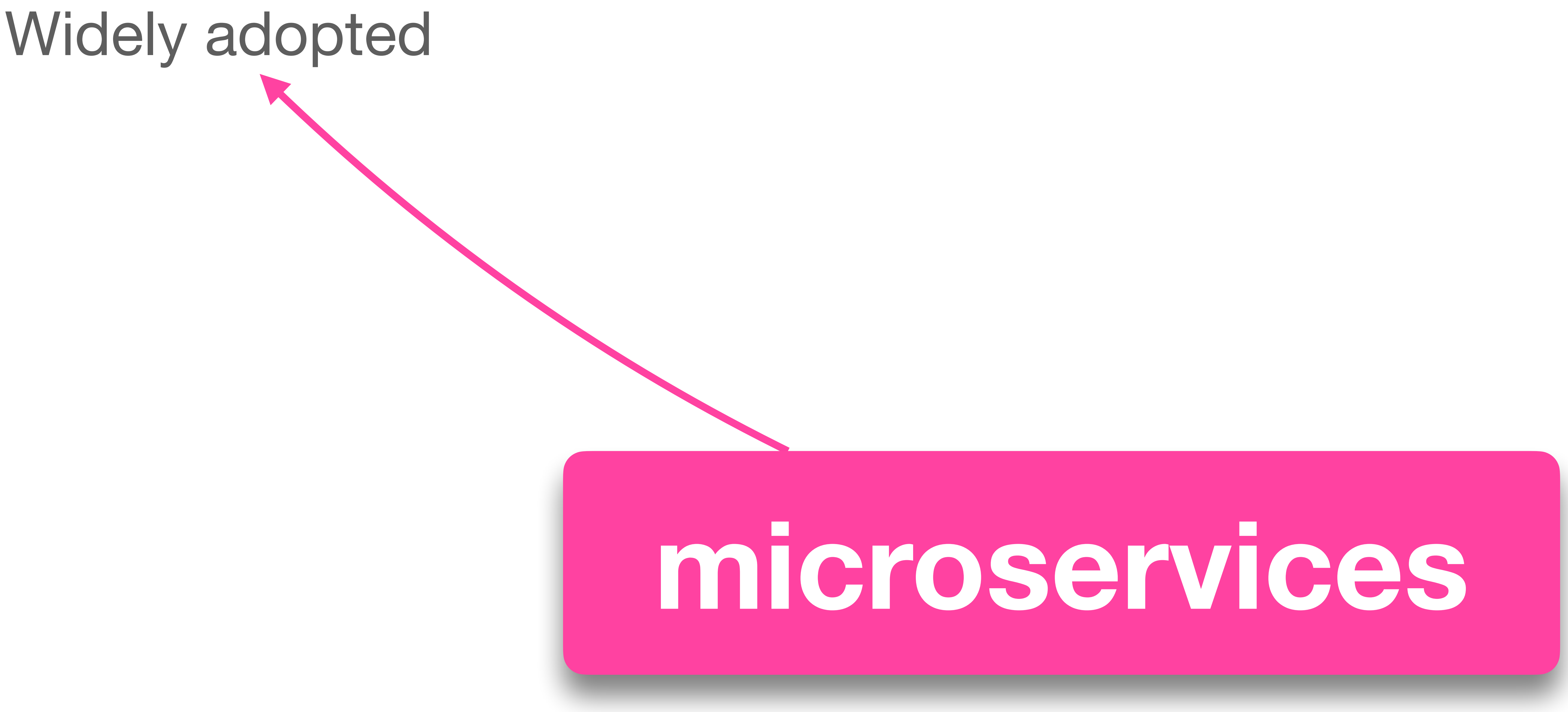
Thatiane de Oliveira Rosa

Alfredo Goldman vel Lejbman

IME

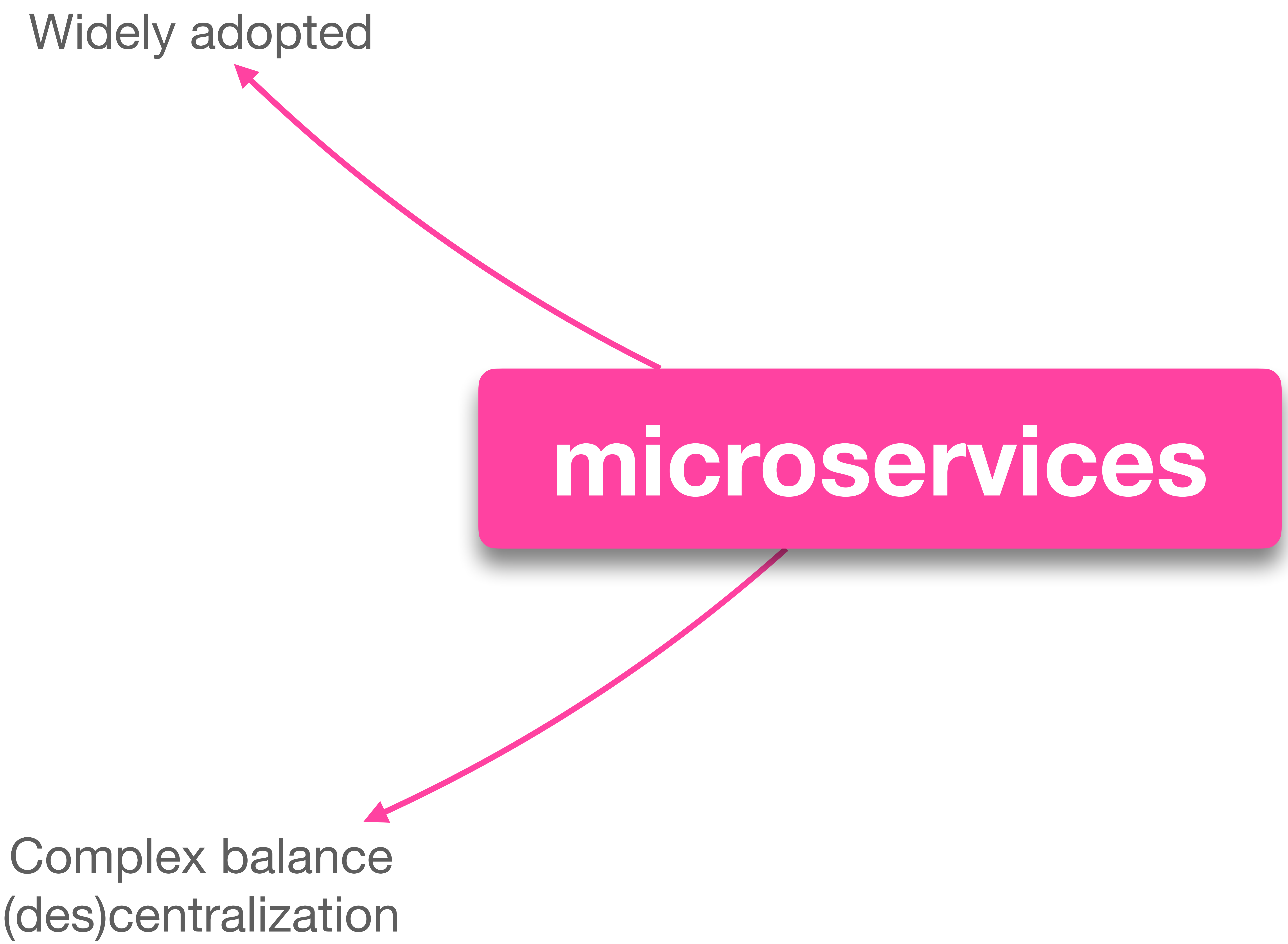
2023

microservices



Widely adopted

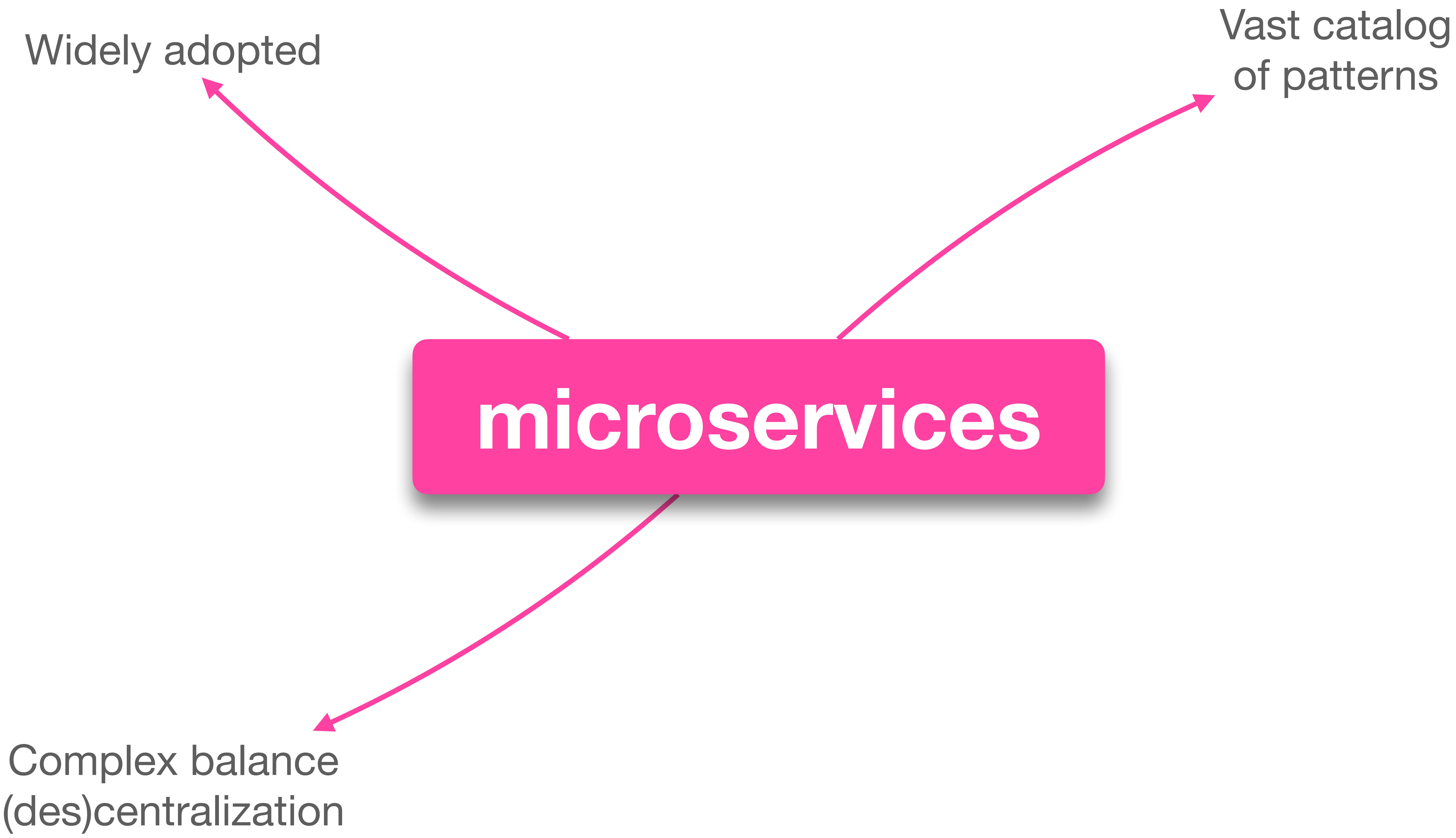
microservices



Widely adopted

microservices

Complex balance
(des)centralization

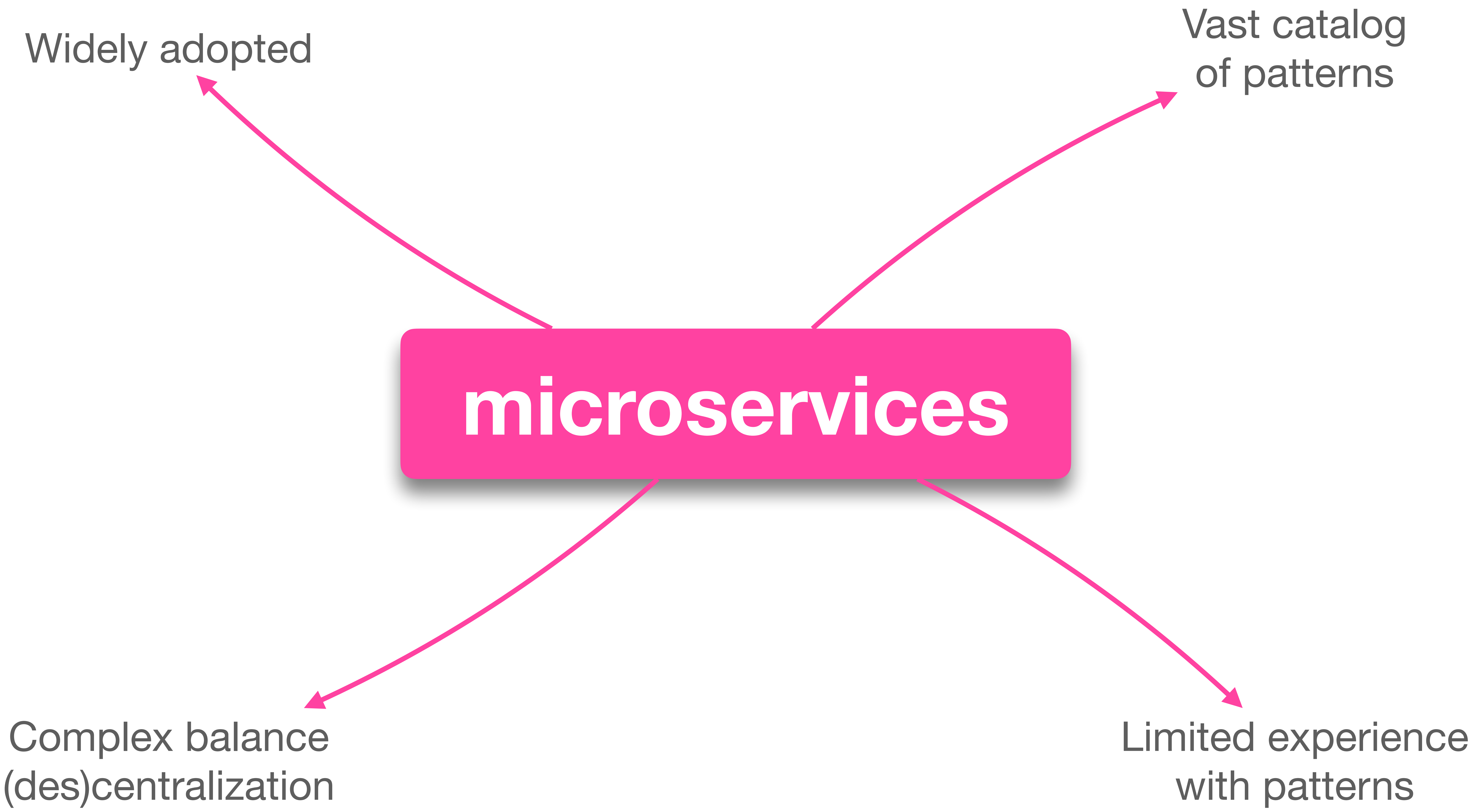


microservices

Widely adopted

Vast catalog of patterns

Complex balance (des)centralization



microservices

Widely adopted

Vast catalog of patterns

Complex balance (des)centralization

Limited experience with patterns

Research Questions

Research Questions

How to increase awareness to MSA patterns agnostic to implementation details?

Research Questions

Can metrics be used to represent MSA patterns? If so, how to combine them?

Research Questions

What are the usages for detection of MSA patterns during the lifecycle of an architecture? And what are the impacts of such usage?

Research Questions

How to increase awareness to MSA patterns agnostic to implementation details?

Can metrics be used to represent MSA patterns? If so, how to combine them?

What are the usages for detection of MSA patterns during the lifecycle of an architecture? And what are the impacts of such usage?

We propose a metric-based approach for detection of MSA patterns

We propose a metric-based approach for detection of MSA patterns

- tech agnostic, wider applicability

We propose a metric-based approach for detection of MSA patterns

- actualised in a open source tool

We propose a metric-based approach for detection of MSA patterns

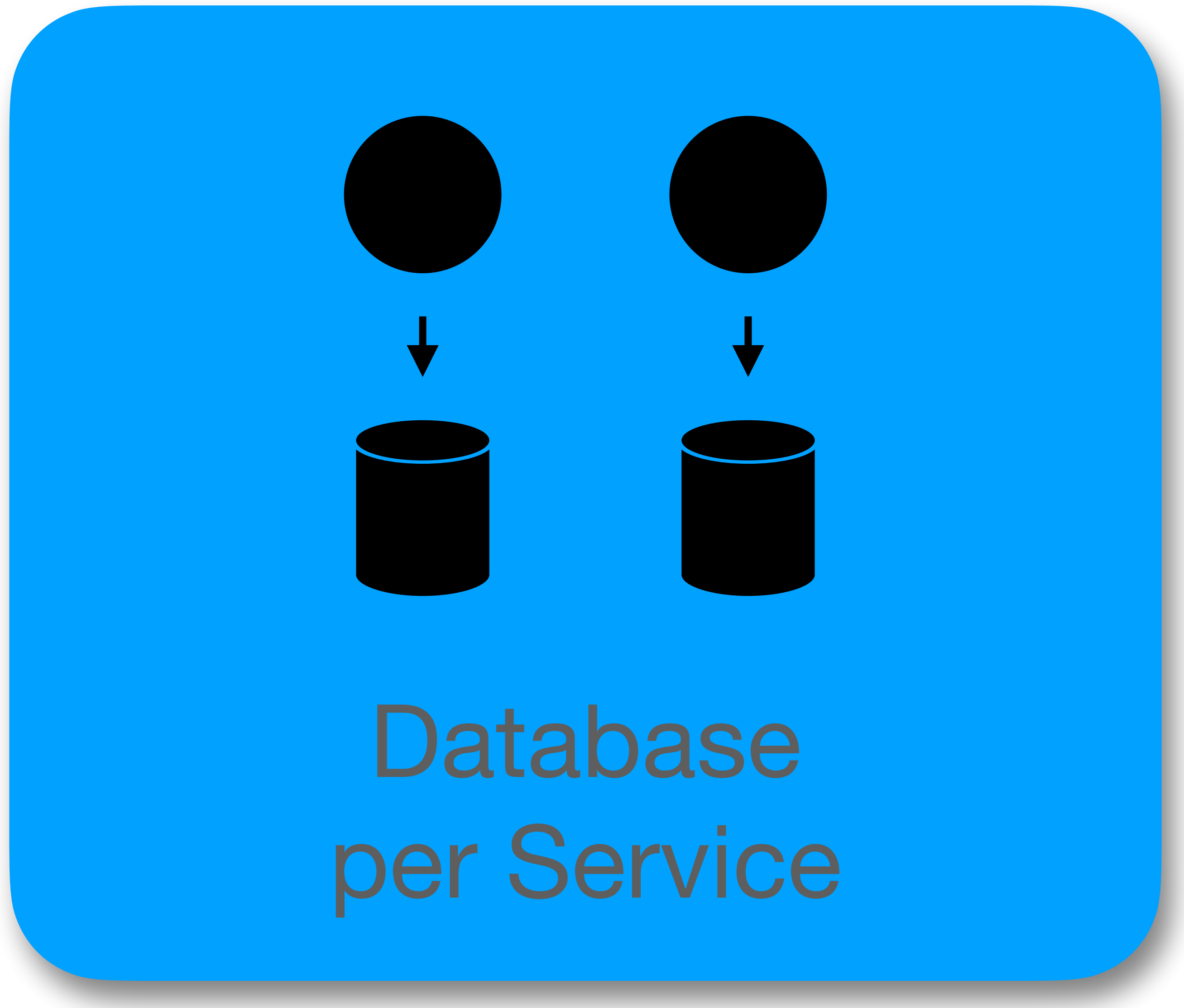
- accuracy and broad utility assessed

We propose a metric-based approach for detection of MSA patterns

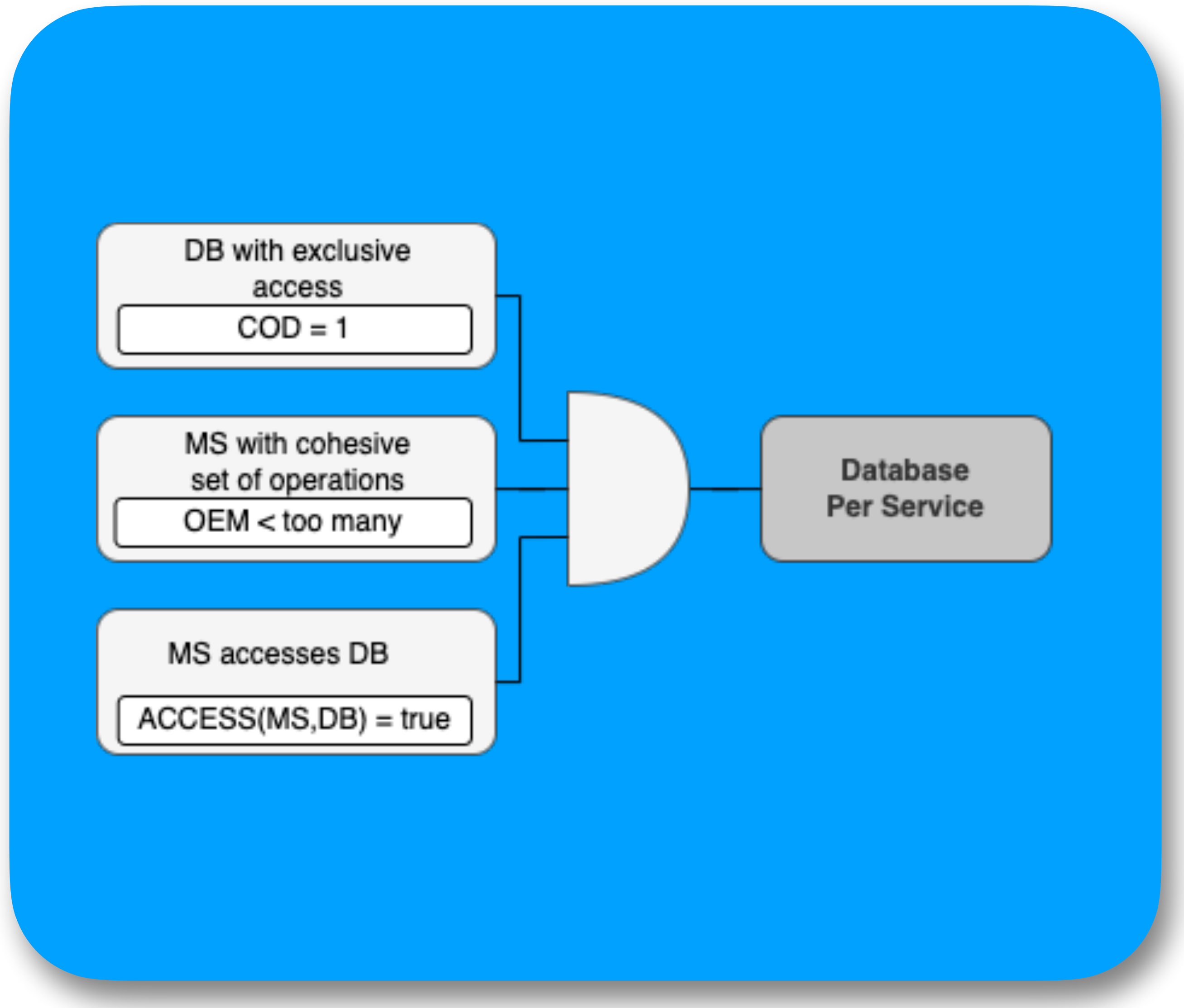
- tech agnostic, wider applicability
- actualised in a open source tool
- accuracy and broad utility assessed

First MSA pattern

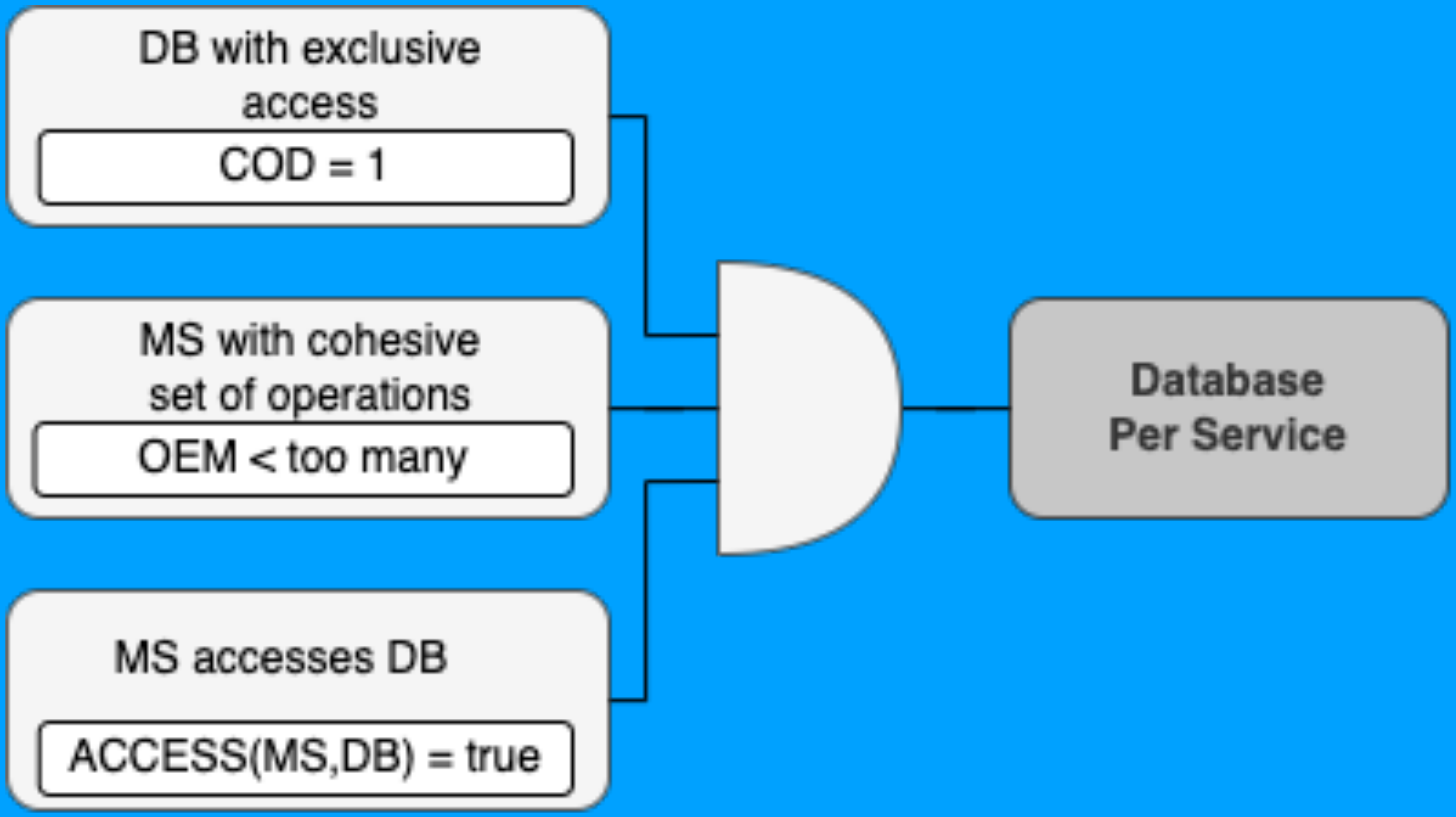
First MSA pattern



Detection Strategy



Det



DB with exclusive access

COD = 1

MS with cohesive set of operations

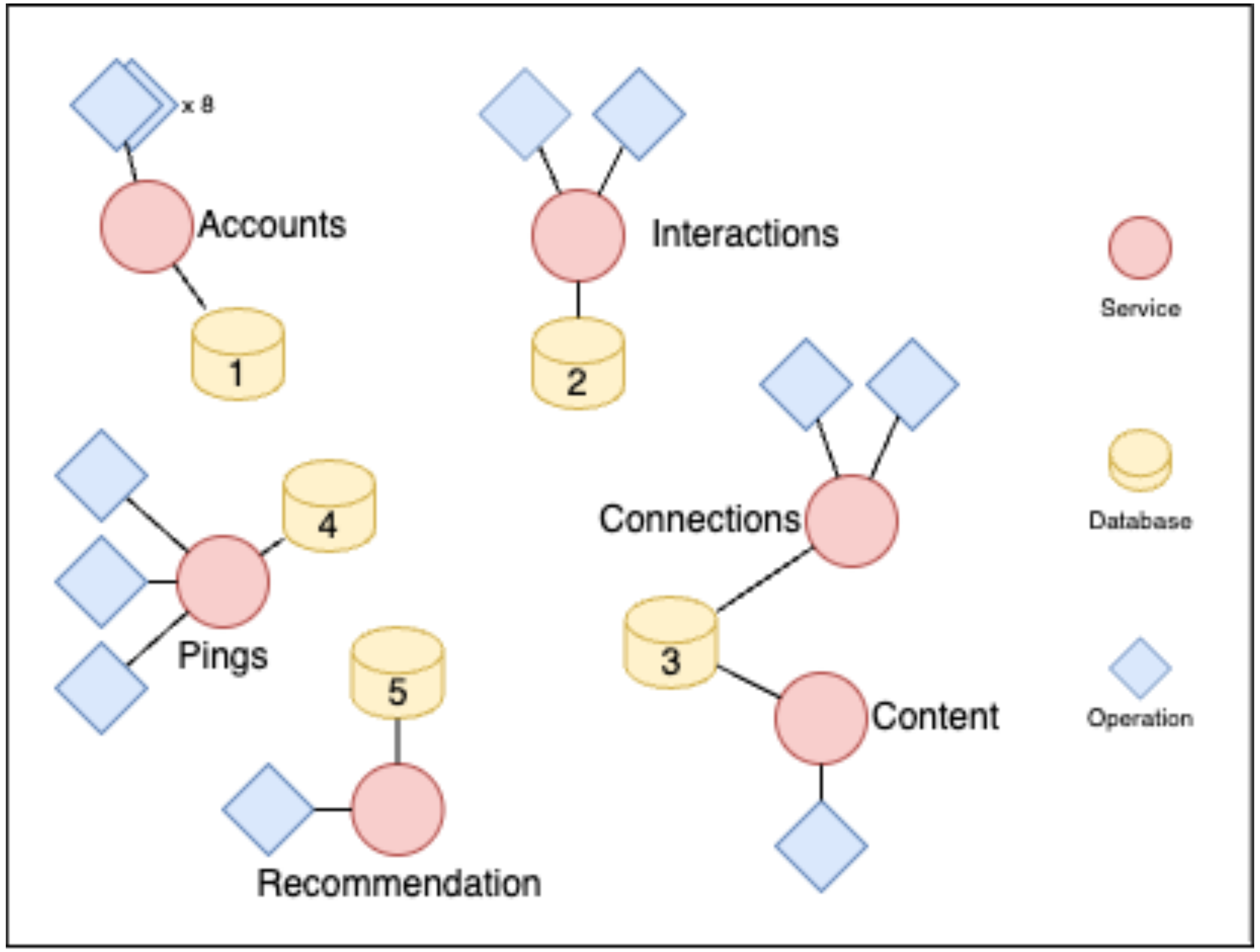
OEM < too many

MS accesses DB

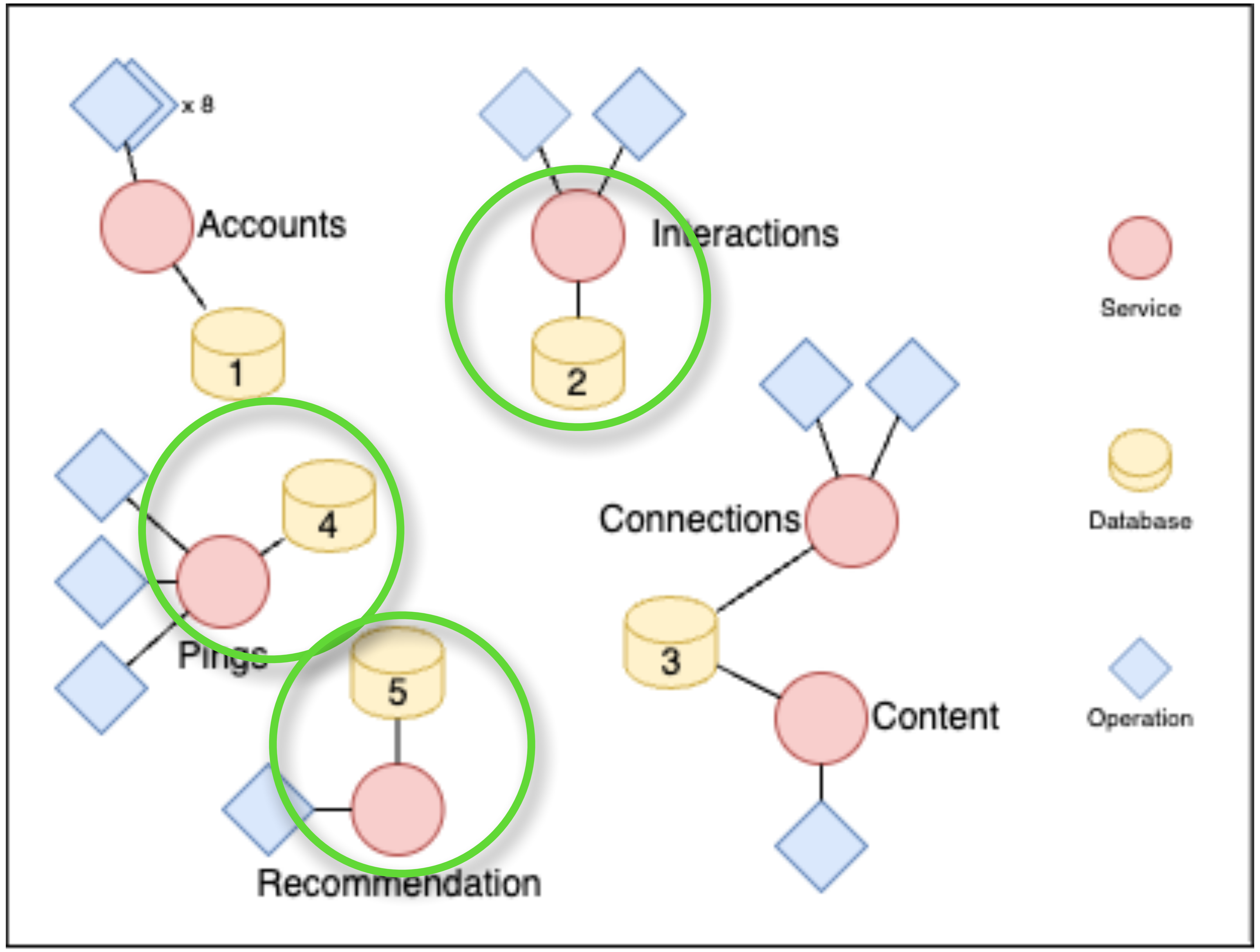
ACCESS(MS,DB) = true

Database Per Service

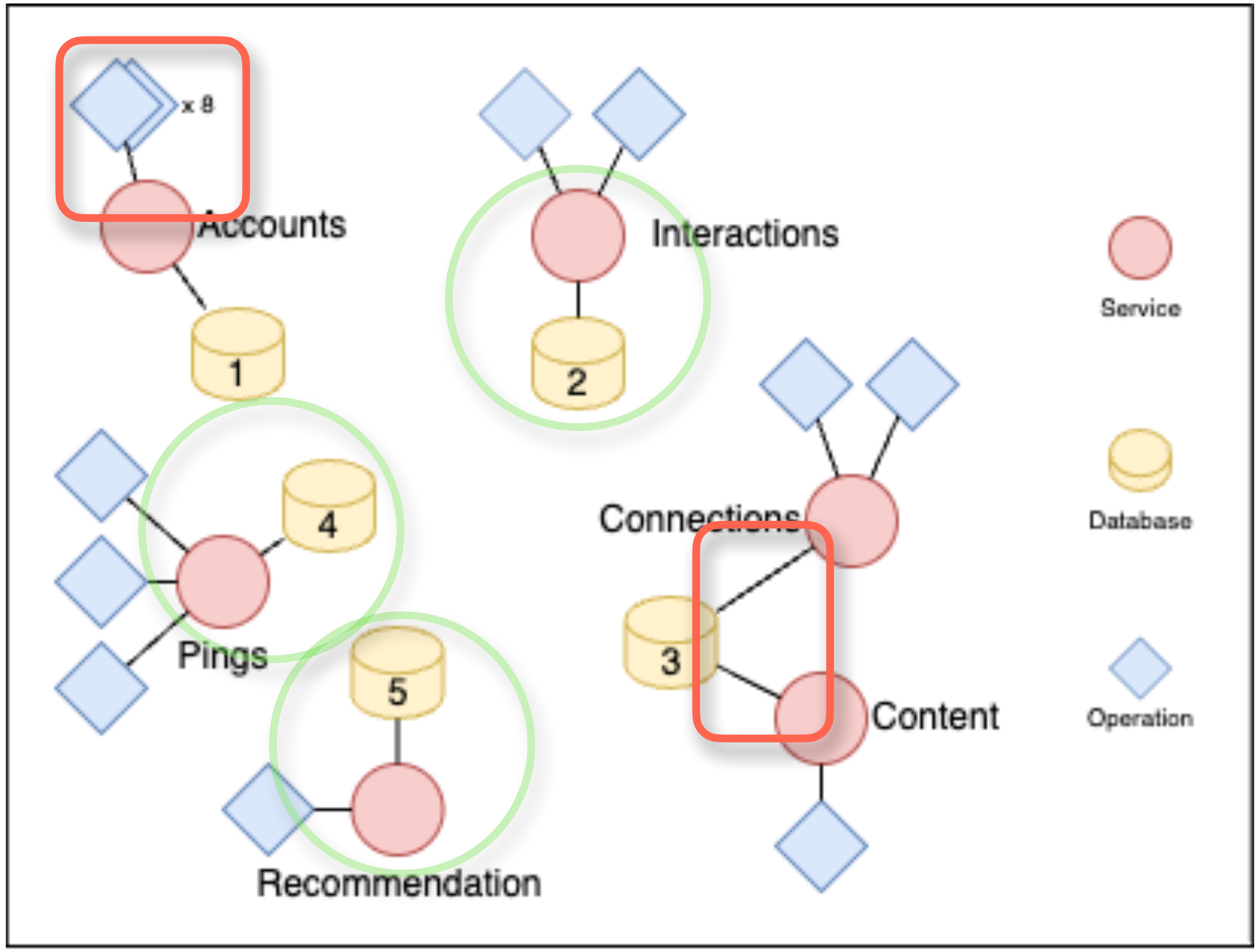
Preliminary Assessment



Preliminary Assessment

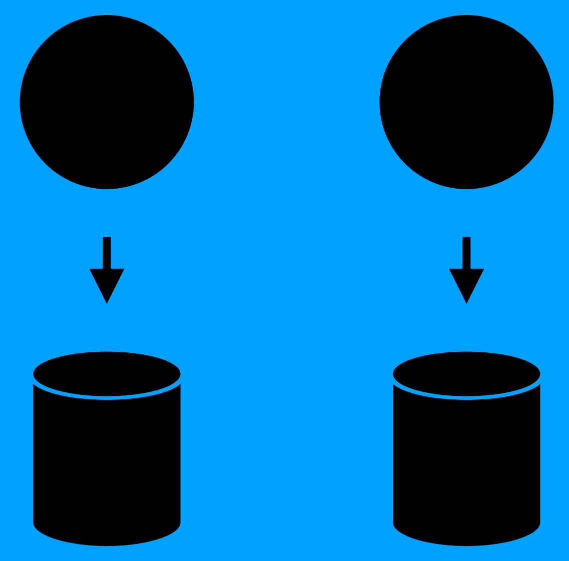


Preliminary Assessment

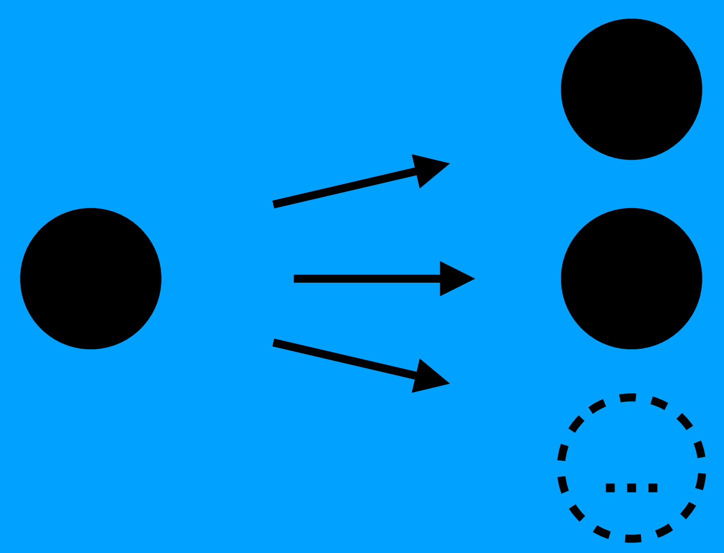


More MSA patterns

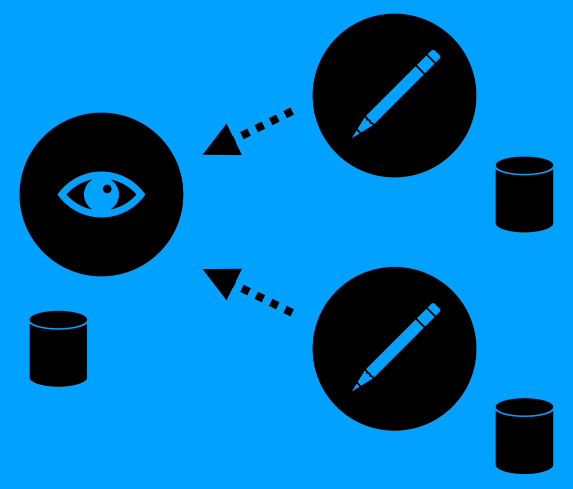
More MSA patterns



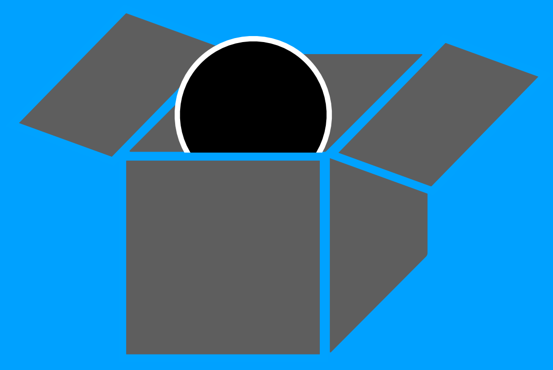
Database per Service



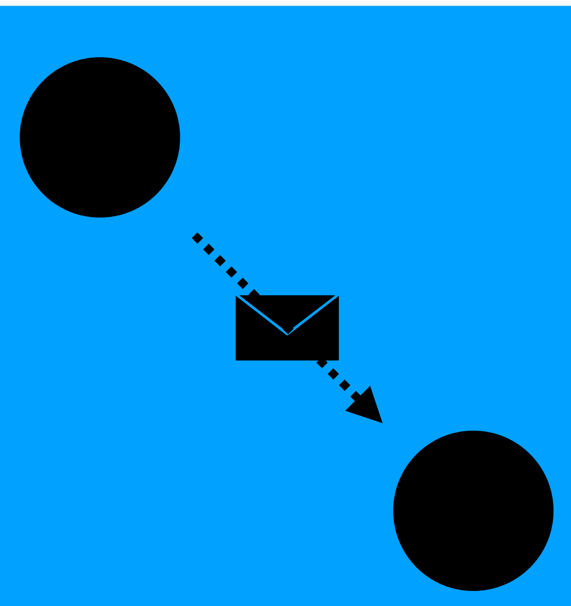
API Composition



CQRS

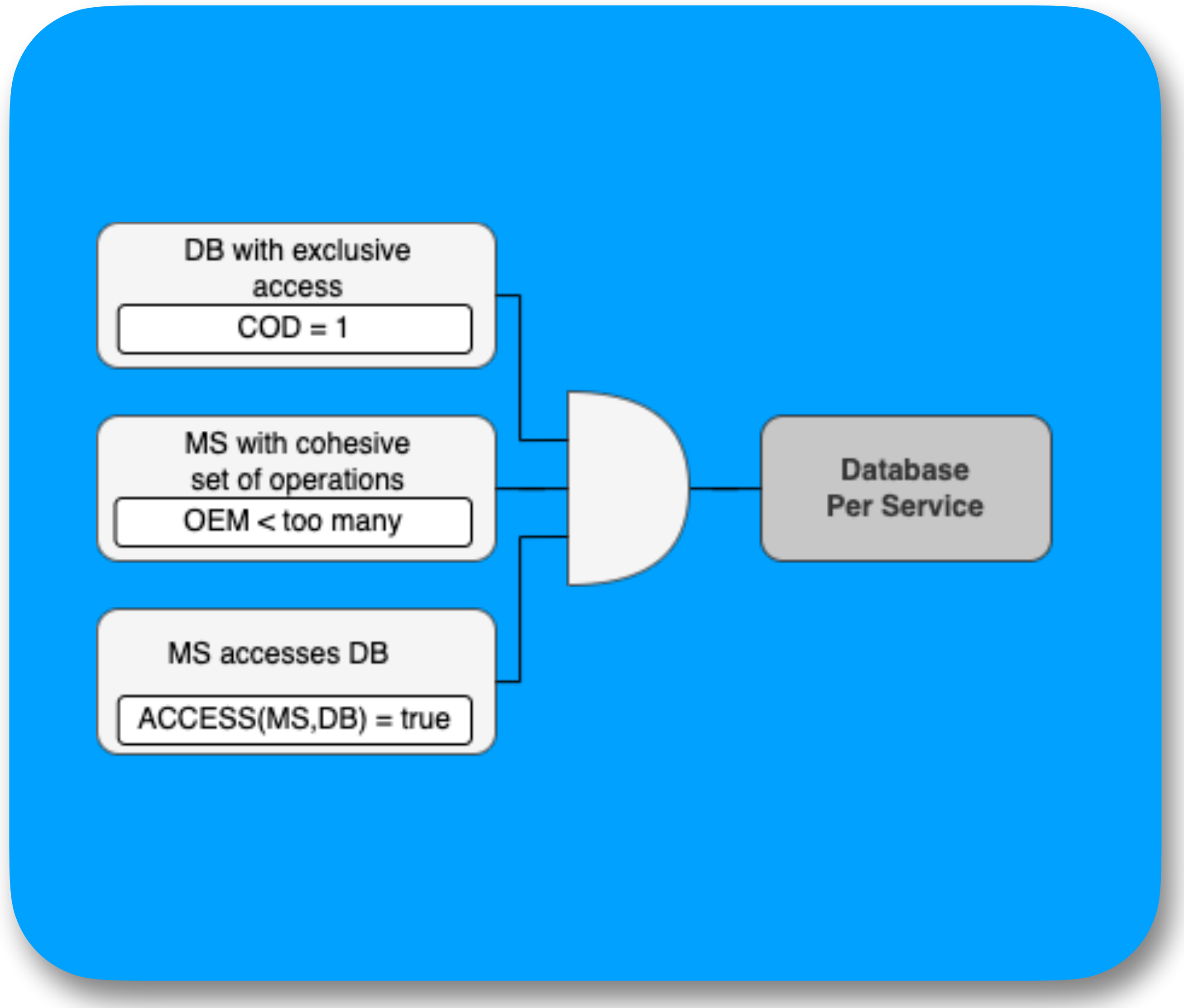


Single Service per Host



Async Message

More detection strategies



DB with exclusive access

COD = 1

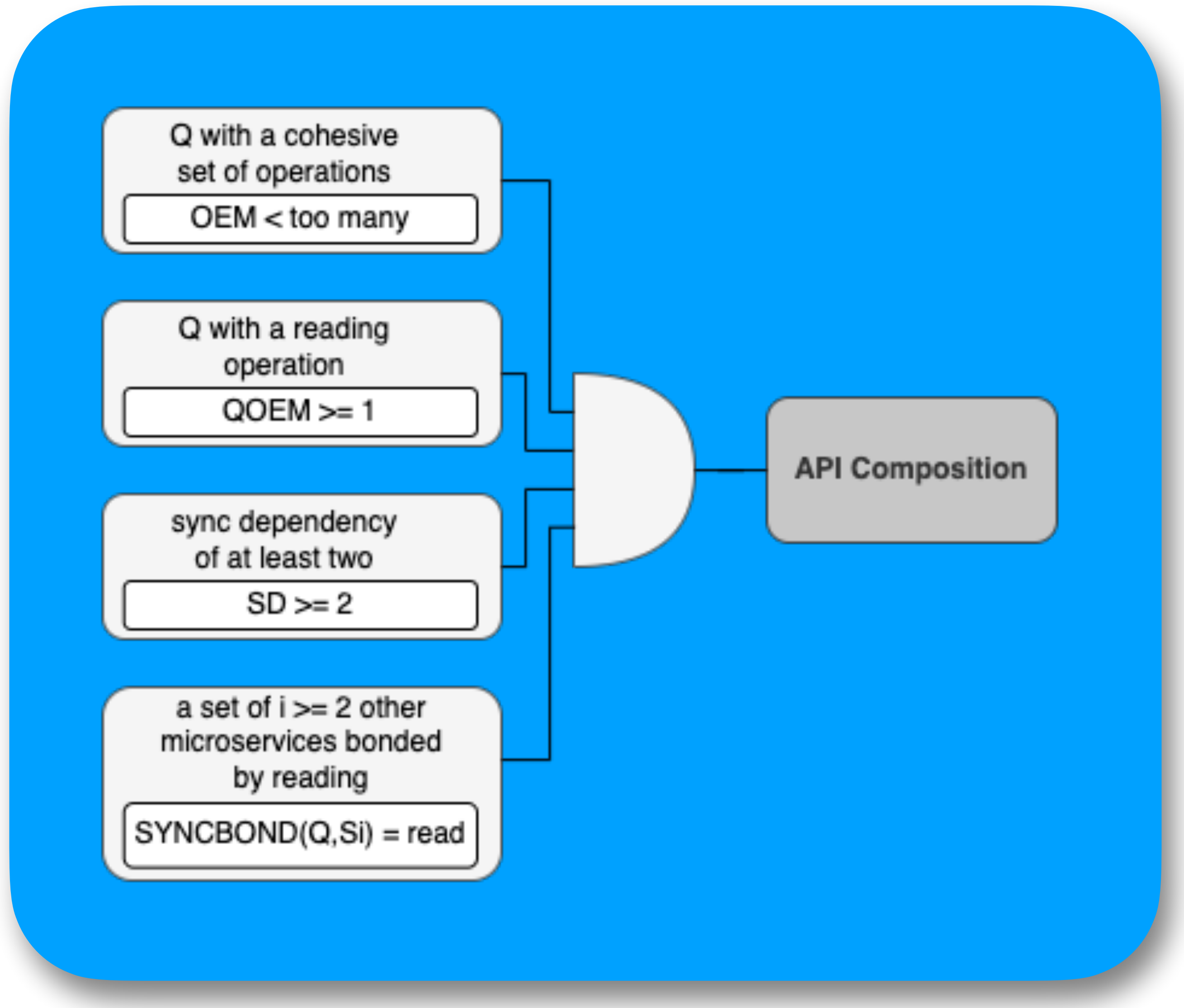
MS with cohesive set of operations

OEM < too many

MS accesses DB

ACCESS(MS,DB) = true

Database Per Service



Q with a cohesive set of operations

OEM < too many

Q with a reading operation

QOEM >= 1

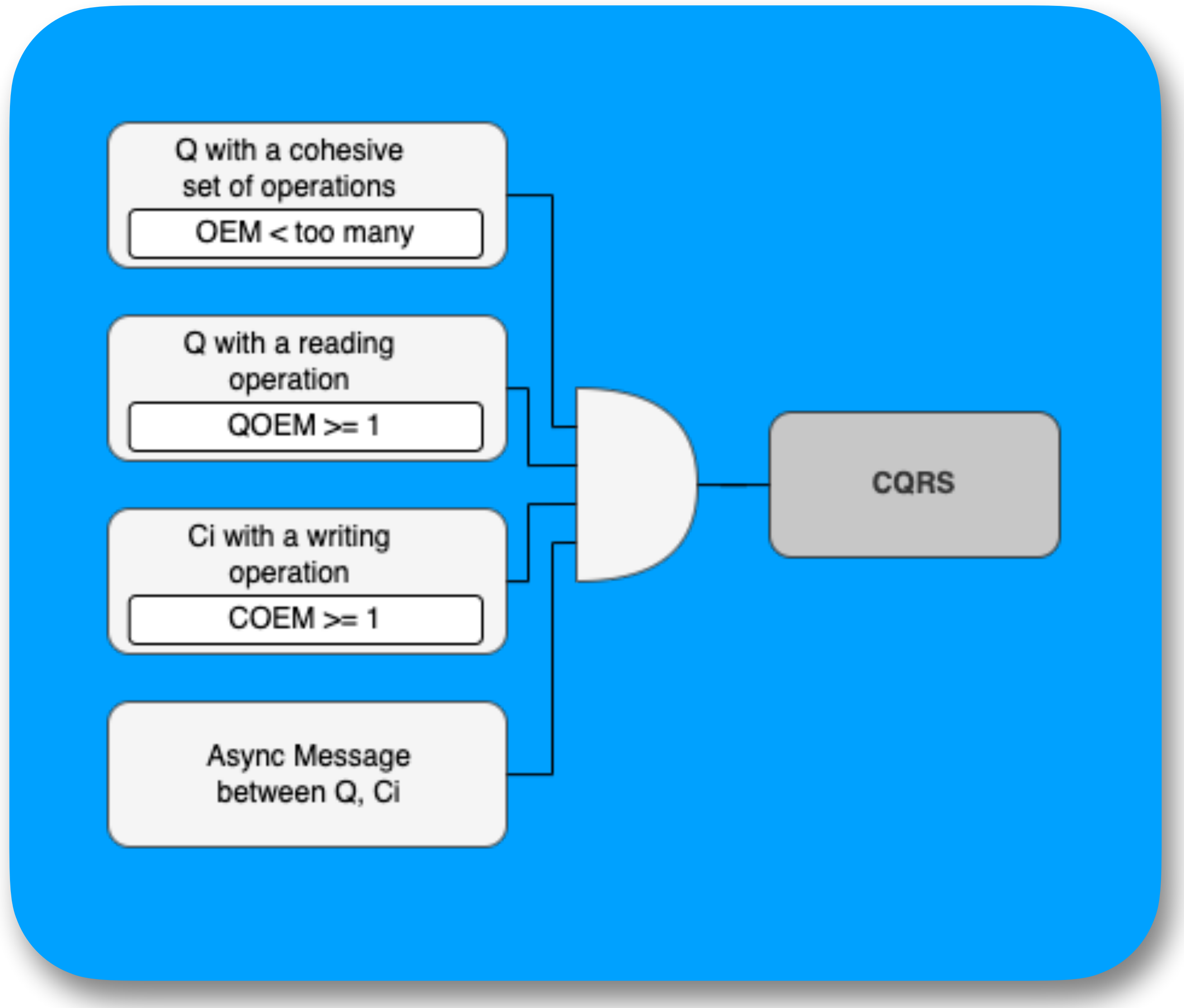
sync dependency of at least two

SD >= 2

a set of i >= 2 other microservices bonded by reading

SYNCBOND(Q,Si) = read

API Composition



Q with a cohesive set of operations

OEM < too many

Q with a reading operation

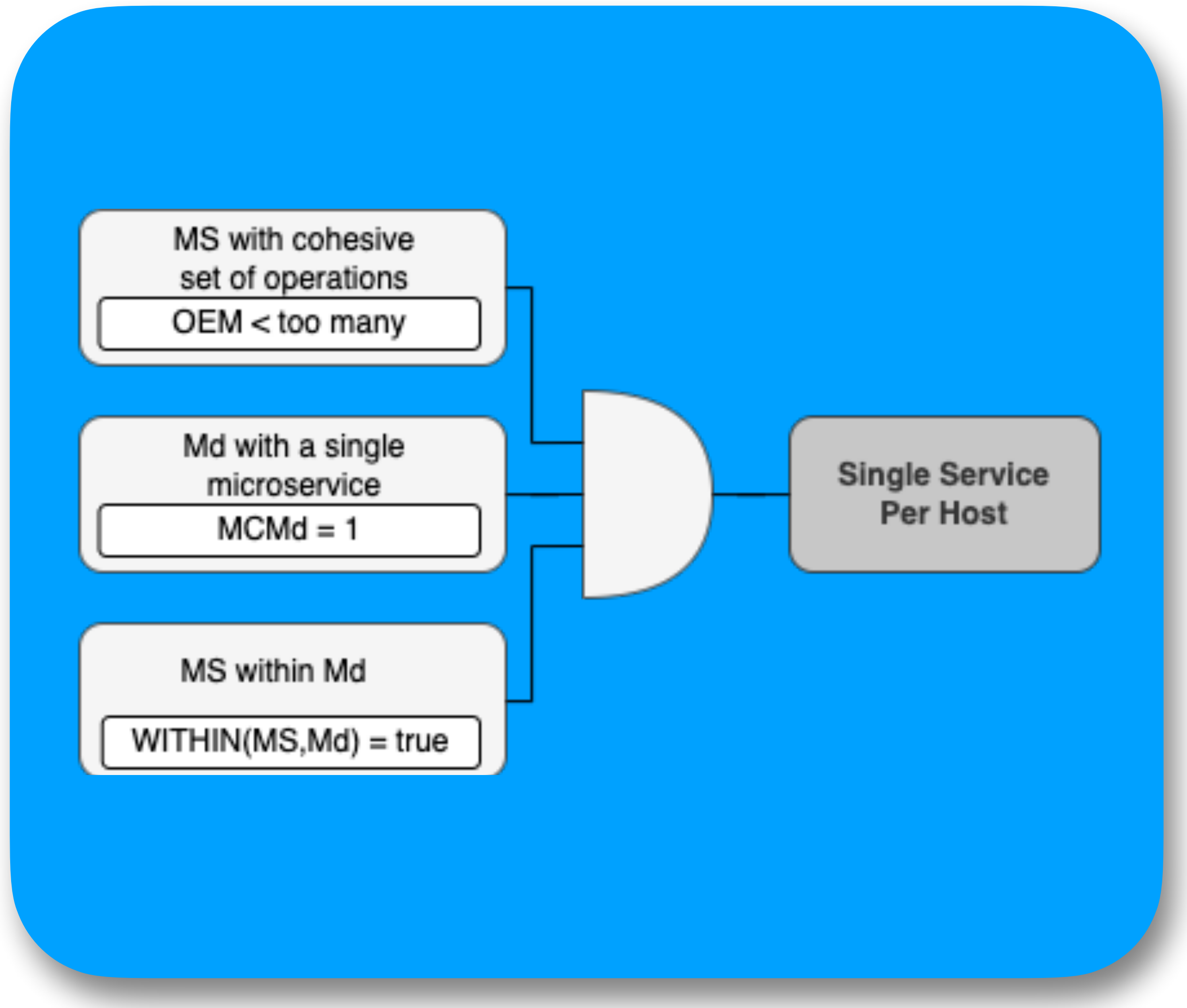
QOEM >= 1

Ci with a writing operation

COEM >= 1

Async Message between Q, Ci

CQRS



MS with cohesive set of operations

OEM < too many

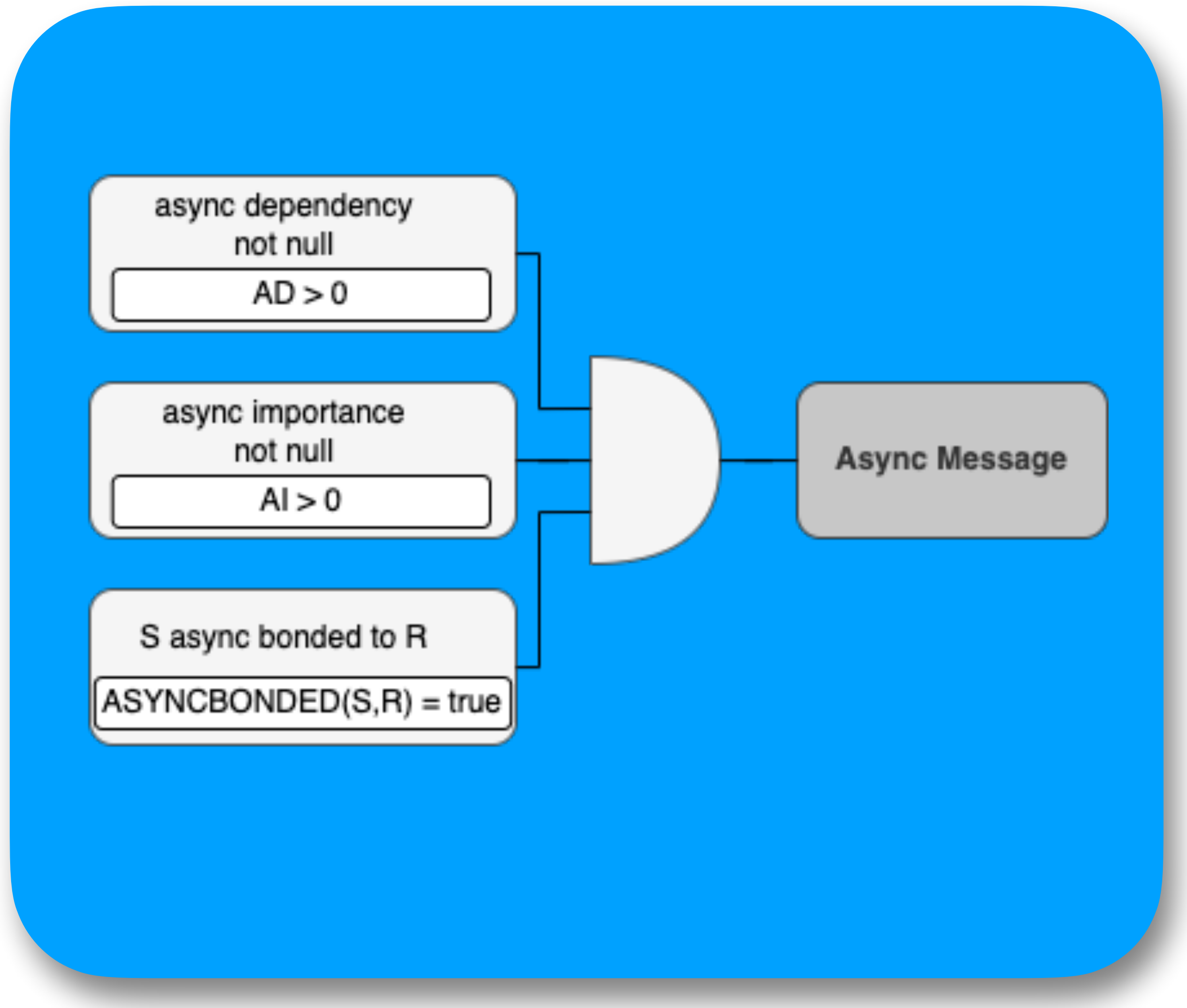
Md with a single microservice

MCMd = 1

MS within Md

WITHIN(MS,Md) = true

Single Service Per Host



async dependency not null

AD > 0

async importance not null

AI > 0

S async bonded to R

ASYNCBONDED(S,R) = true

Async Message

Cases study

Cases study

4	3	0	8	1
Database per Service	Single Service per Host	API Composition	Async Message	CQRS

InterSCity

Cases study

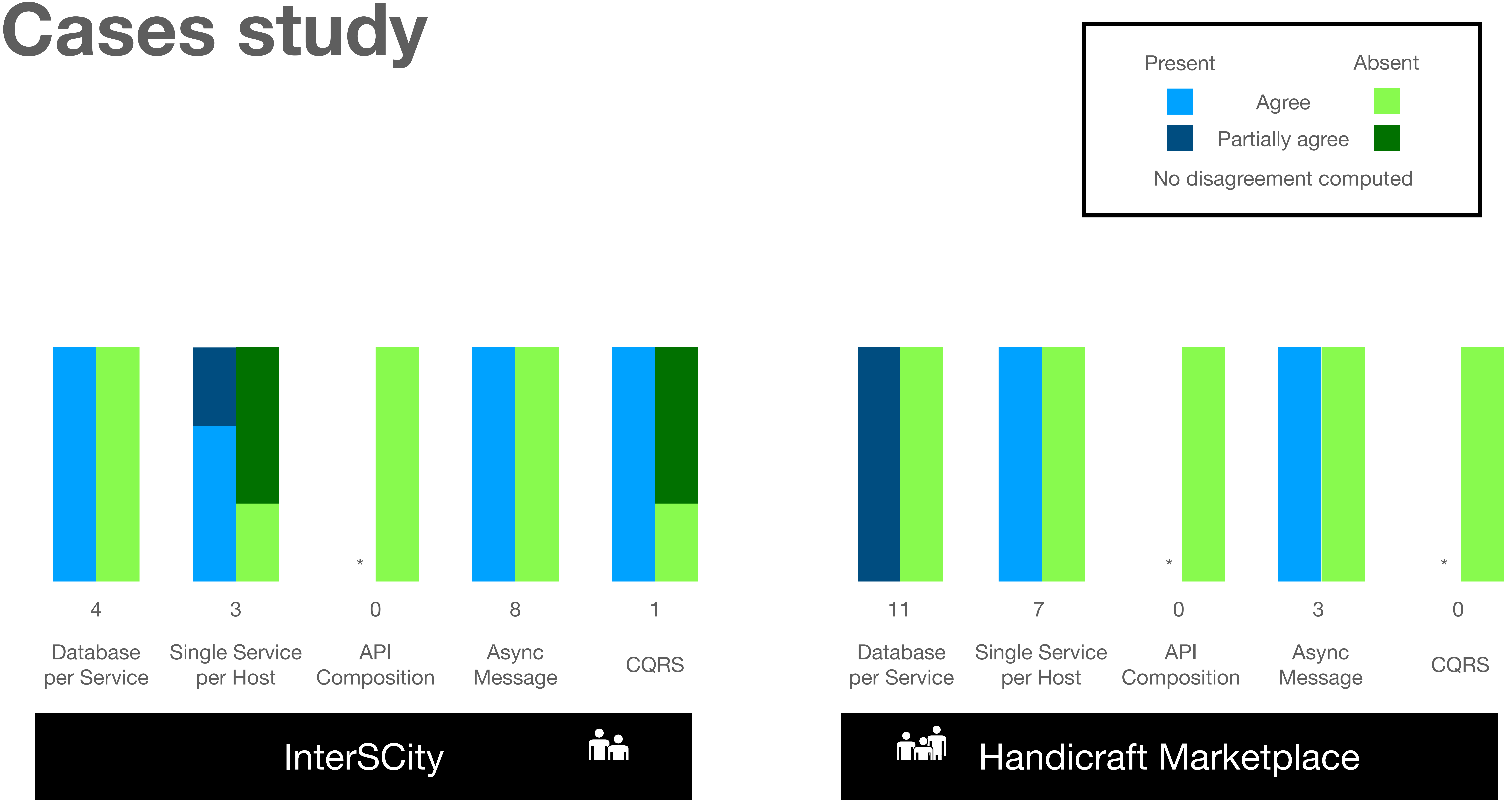
4	3	0	8	1
Database per Service	Single Service per Host	API Composition	Async Message	CQRS

InterSCity

11	7	0	3	0
Database per Service	Single Service per Host	API Composition	Async Message	CQRS

Handicraft Marketplace

Cases study


Present
Absent
Agree
Partially agree
No disagreement computed
*
*
*
4
3
0
8
1
Database per Service
Single Service per Host
API Composition
Async Message
CQRS
InterSCity
11
7
0
3
0
Database per Service
Single Service per Host
API Composition
Async Message
CQRS
Handicraft Marketplace

Take-aways

Accuracy

Take-aways

Accuracy

Limitations

How can usVision be sure that [one of the microservices] is performing the join internally?

Take-aways

Accuracy

Limitations

I agree with all the detections, and I think there should be more occurrences (…) something we need to implement yet

Invaluable

The original goal of [one of the microservices] was to act as the composer element of this pattern, but from what I see here, there might have been a drift during the implementation

Take-aways

Accuracy

Limitations

Invaluable

Improvements

Towards the Detection of Microservice Patterns Based on Metrics

Thank you and please reach out

✉ joao.daniel@student.unibz.it

João Francisco Lino Daniel

2023