From monolithic to microservice architecture: an automated approach based on graph clustering and combinatorial optimization

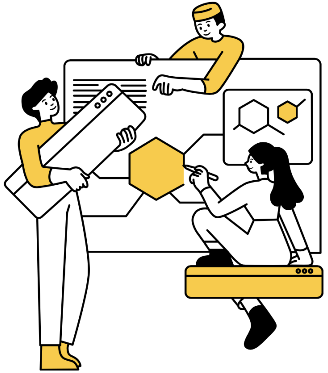
Gianluca Filippone, Nadeem Qaisar Mehmood, Marco Autili, Fabrizio Rossi, Massimo Tivoli

Università degli Studi dell’Aquila

Dipartimento di ingegneria, scienze dell'informazione e matematica

Challenges when migrating to microservices

- High-cohesive and low coupled microservices
- Right granularity
- Bounded contexts
- Single-responsibility principle



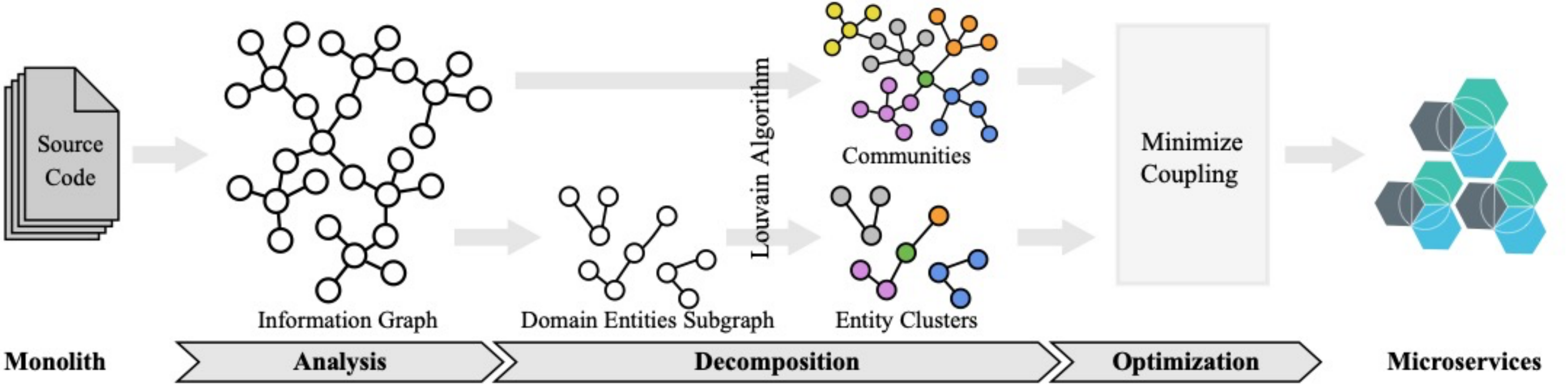
Complex, time consuming, error prone...

Goals

- Process automation
- No manual input
- No specific skills or system knowledge

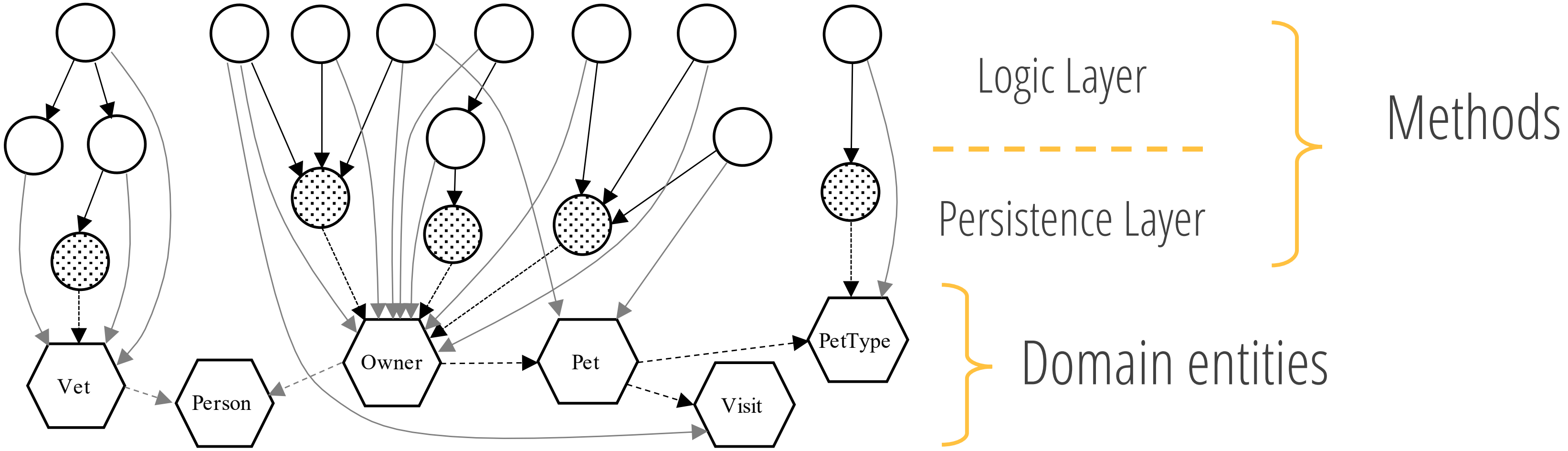
- Focus on cohesion and coupling
- Bounded contexts

Approach overview



Static Code Analysis

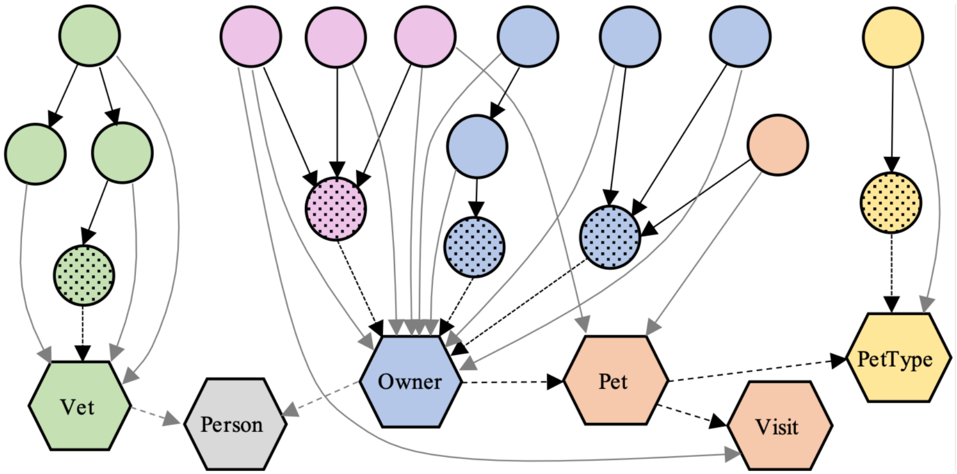
- Layered architecture
- Directed weighted graph
  - **Nodes:** methods and domain entities
  - **Arcs:** code relationships



System decomposition (1)

- Microservice extraction → community detection problem

- **Goal**: communities of densely linked nodes
- **Method**: Louvain clustering algorithm



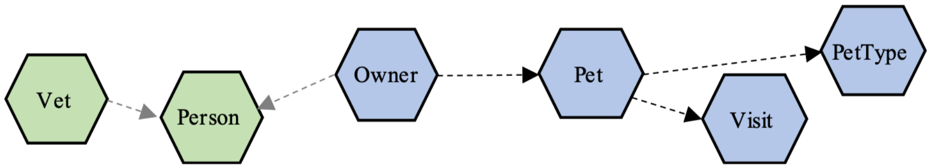
- Cohesive communities
- Nodes from all layers

- No guarantees over the structure
- No guarantees over coupling

System decomposition (2)

Clustering applied:

1) On the whole graph *(communities)*
2) On the entities subgraph *(entity clusters)*
   - Nodes of type Entity
   - Relationships with the persistence layer



- Identify service granularity
- Domain driven

Optimization (1)

- **Goal:** partition the graph into *m* sets (microservices) to minimize coupling
- **Method:** ILP formulation of a variant of the Multiway Cut problem

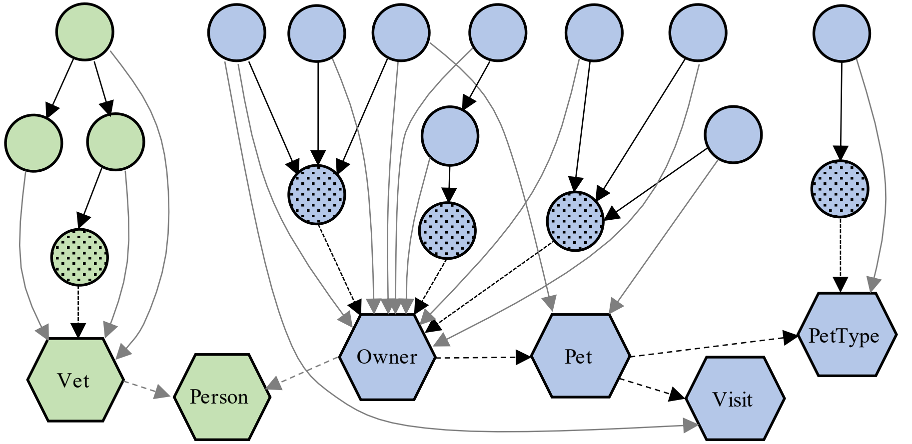
Problem statement

Given a graph $G = (V, E)$, a type $t_i \in \{Method, Entity\}$ for each node $i \in V$, a weight w_{ij} for each edge $(i, j) \in E$, **find** a partition of V in m sets $\{M_1, \dots, M_m\} = M$, such that:

i. $\bigcap M_i = \emptyset$
ii. $\bigcup M_i = V$
iii. Each $M_k \in M$ contains at least one node i s.t. $t_i \neq Entity$
iv. The sum of weights w_{ij} s.t. $i \in M_h,\ j \in M_k,\ h \neq k$ is minimized

Optimization (2)

- Low coupling, but not domain-driven decomposition
- Communities from previous phase:
  - Distribute entities into microservices
  - Nodes from same communities in the same microservice
  - Fix-and-relax approach



- Vertical decomposition
- Autonomous services
- Single responsibility
- Bounded to contexts

Evaluation: goals and experiments

Goals

- Have a well modularized architecture (high cohesion and low coupling)
- Follow single responsibility principle
- Independently offer functionality
- Built around the right bounded contexts

Experiments

- Publicly available monolithic systems
- Comparison of results with other sota approaches

Evaluation: results (1)

- Project: JPetStore

Approach	# Microservices	Average Coupling	Average Cohesion	IFN	# Method calls	# Entity references
Baseline	3	1.26	0.95	2.0	3	2
Zaragoza et al.	3	3.13	0.90	2.0	6	7
Selmadji et al.	2	3.8	0.93	2.5	3	8
Jin et al.	4	2.25	0.82	1.75	7	5
Brito et al.	4	3.1	0.85	1.75	9	8
Our	3	<u>0.87</u>	<u>0.97</u>	<u>1.7</u>	<u>3</u>	<u>0*</u>

* Optimal value

Evaluation: results (2)

- **Project:** Spring PetClinic

Approach	# Microservices	Average Coupling	Average Cohesion	IFN	# Method calls	# Entity references
Baseline	3	1.2	0.75	1.7	2	3
Kamimura et al.	3	2.07	0.76	1.7	2	7
Our	2	0.0*	1*	2.0	0*	0*

* Optimal value

- **Project:** Cargo Tracking System
  - 3 Microservices (3 bounded contexts)
  - Precision: 0.94, Recall: 0.83

Final remarks on evaluation results

- Fully automatic approach

- High cohesion and loose coupling
- Independent and self-contained functionalities
- Bounded-context

Encouraging results for a promising approach

Threats to validity

- Real-scale projects
- Pre-defined weights not always best-fitting
- Quality of results VS system complexity

- Customizable weights and clusters
- Approach can be extended to dynamic analysis

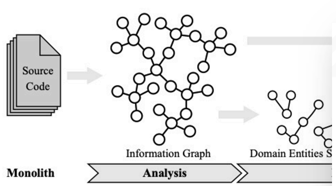
Conclusions

Goals

- Process automation
- No manual input (just source code)
- No specific skills or system kno

- Focus on cohesion and couplin
- Focus on functionalities (and d

Approach overview



Evaluation: results (1)

- Project: JPetStore

Approach	# Microservices	Average Coupling
Baseline	3	1.26
Zaragoza et al.	3	3.13
Selmadji et al.	2	3.8
Jin et al.	4	2.25
Brito et al.	4	3.1
Our	3	0.87

* Optimal value

Final remarks on evaluation results

- Fully automatic approach

- High cohesion and loose coupling
- Independent and self-contained functionalities
- Bounded-context Microservices

Encouraging results for a promising approach

Thank you!

Evaluation metrics

- Average coupling

$$\frac{1}{|M|}\sum_{(i,j)\in E|i\in M_h,j\in M_k,h\neq k} w_{ij}$$

- Average cohesion

$$\frac{1}{|M|}\sum_{M_k\in M}\frac{inner_k}{outer_k}$$

- IFN

$$\frac{1}{|M|}\sum_{M_k\in M} ifn_k$$

- Precision and Recall

$$Precision(k,d) = \frac{|E_d^k \bigcap E_B^k|}{|E_d^k|}$$

$$Recall(k,d) = \frac{|E_B^k|}{|E_d^k \bigcap E_B^k|}$$