DYNAMOS
Dynamically Adaptive Microservice-based OS
A Middleware for Data Exchange Systems

Jorrit Stutterheim
Ana Oprescu
Thomas van Binsbergen
Table of contents

01 Background
Digital Data-sharing Marketplaces

02 Method
Use case and research questions

03 DYNAMOS
Core concept

04 Research
Developing DYNAMOS

05 Discussion
Experiment results

06 Conclusion
Concluding remarks and future work
01

Background

Digital Data-sharing Marketplaces
Amsterdam Data Exchange (AMdEX)

**Consortium**
Of partners that are working on establishing a ‘Digital Data-sharing Marketplace’

**Digital Data-sharing Marketplace**
That enables participating organizations to maintain control over their data and enforce access and control through legal contracts, called policies

**Funding**
AMdEX funded part of this thesis
Digital **Data-sharing** Marketplace
Data-exchange patterns or archetypes

02

Method

Use case and research questions
“This research focuses on **abstracting** different **data exchange patterns** into a unified set of core data exchange **microservices** that adhere to agreed-upon data exchange **policies**”
The **Why**

**Flexibility**
Compose lightweight microservices in different patterns to offer data-exchange services that adhere to policy agreements

**Decoupling development**
Develop and deploy data-exchange microservices without touching the core system

**Green IT**
Scalability and flexibility of microservices, will allow to optimize for resource consumption, saving energy
Method

**AMdEX use case**
A research data-exchange case, for the universities of the Netherlands (UNL) to derive requirements

**Action research**
Develop a prototype to analyze methods, advantages and disadvantages of this microservice approach

**Literature**
Research defines archetypes and proposes possible architectures and how to adopt self-adaptability
Use Case

Meet Ana Lyst
data scientist

*Pictures made by storyset.com*
Wage gap analysis on Dutch universities
Use AMdEX
To pick and analyze datasets
Perform analysis
Without seeing the data
Archetype compute to data
Archetype data through a trusted third-party
Microservice design

- Query the dataset
- Anonymize sensitive columns
- Aggregate the results of two or more queries
- Algorithm: Perform the analysis
- Graph: Convert results to a graph
Research questions

What are the advantages and disadvantages of different approaches to automatically re-compose a running microservices architecture in a distributed system?

What attributes/properties should we have to compose microservices in a DDM environment?

What extra attributes/properties should we have to make decisions on extra-functional properties, like energy consumption and performance?
DYNAMOS
Dynamically Adaptive Microservice-based OS
“Inspired by the way traditional Operating Systems abstract the complexities of computer architectures into **standardized core functions**

DYNAMOS **abstracts** the **complexity of data-exchange** systems, into lightweight microservices”

**Distributed system**

With OS-like properties
## (Some) key distributed OS properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transparency</strong></td>
<td>Backend processing is transparent (invisible) to Ana</td>
</tr>
<tr>
<td><strong>Concurrency</strong></td>
<td>Distributed parts of the system process requests concurrently</td>
</tr>
<tr>
<td><strong>Resource Management</strong></td>
<td>System load is evenly distributed</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>System can scale to accommodate load</td>
</tr>
<tr>
<td><strong>Flexibility</strong></td>
<td>The best targets for a type of computation can be selected</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>Processes do not interfere with each other, unauthorized access is forbidden</td>
</tr>
</tbody>
</table>
Architecture in literature

Theoretical architecture

Key terms
- Request handler
- Policy Matching module

Relevant concept
- **Request handler**
  Ensures requests are processed
- **Policy matcher**
  Checks policy for what is allowed

DYNAMOS
- Seen as a distributed operating system
- Distributed agents, informed by an orchestrator
- Policy enforcer to validate access
Abstract architecture
Communication Abstraction

- DYNAMOS internal communication
- User initiated communication

Data stewards

Distributed agent

- Microservices
- Data pod

Orchestrator

- Policy Enforcer

Data store
Sidecar Pattern
Communication Abstraction

**Advantages**

- The main service, does not need to know anything about the communication solution
- There is a written specification, the protocol buffer, which defines the interface
- Type safety
- All services, even in other languages, use the exact same way to communicate

**Disadvantages**

- Latency
- Complexity
Research
Current challenges

01 How to combine microservices and archetypes
02 How to execute the microservices
03 How to transfer data between services
04 How to dynamically select archetypes
How to **combine** microservices and **archetypes**
Microservice **design**

- **Query** the dataset
- **Anonymize** sensitive columns
- **Aggregate** the results of two or more queries
- **Algorithm** Perform the analysis
- **Graph** Convert results to a graph

---

**Data provider**

**Compute provider**
01

How to **combine** microservices and **archetypes**

- Microservice Chain
- Labeled Directed Acyclic Graph
Microservice **Chain** in different **archetypes**

- **Compute to data**
  - Query
  - Anonymize
  - Algorithm
  - Graph

- **Data through Ttp**
  - Query
  - Anonymize
  - Algorithm
  - Graph
  - Aggregate
  - Algorithm
  - Graph
Assign Roles

Role: 'All'

Role: 'Data provider'

Role: 'Compute provider'
98,300,000

Big numbers catch your audience’s attention
Current challenges

01 How to combine microservices and archetypes
02 How to execute the microservices
03 How to transfer data between services
04 How to dynamically select archetypes

Challenges
02
How to **execute** the microservice chains

**Persistent services**
- Constantly running microservices
- Determine request route with the microservice chain
- Use ‘regular’ methods of passing data (AMQP)

**Ephemeral jobs**
- Single use self-cleaning jobs
- Jobs are created with required microservices from the chain
- Custom method of passing data

Both are supported in DYNAMOS
02
How to **execute** the microservice chains

**Persistent services**

**Advantages**
- Readiness, always on and ready to handle requests
- Consistent performance

**Disadvantages**
- Potential security risk: prolonged attack exposure
- Lack of isolation, users will share the same service
- Constant resource consumption

**Ephemeral jobs**

**Advantages**
- Complete isolation
- Fresh state
- Limited exposure time

**Disadvantages**
- Cold starts
- Complexity
- Resource spikes
Current **challenges**

01 How to combine microservices and archetypes

02 How to execute the microservices

03 How to transfer data between services

04 How to dynamically select archetypes
How to transfer data through the microservice chain

DYNAMOS internal communication
User initiated communication
How to **transfer** data through the microservice chain

**Persistent services**

- Sidecar pattern to abstract away communication
- Kubernetes environment
- Agent manages input and output
- The microservice chain is stored in the message
How to **transfer** data through the microservice chain

**Ephemeral jobs**

- Inter Process Communication (IPC)
- Happens within a pod
- Sidecar and agent for input/output
Current challenges

01 How to combine microservices and archetypes
02 How to execute the microservices
03 How to transfer data between services
04 How to dynamically select archetypes
Why **change** archetypes?

- Change in policy agreement
- Aggregation of results on a third party
- Offload processing to third parties if the system is under stress

**Self-adaptivity**
MAPE-K feedback loop

*Palande, S. S.  
Situation-Aware Self-Adaptive Localisation Framework
How to dynamically select archetypes
Configuration management
In a distributed system

“Distributed applications require different forms of coordination. Configuration is one of the most basic forms of coordination”

Configuration management

- Distributed system state
- Static Configuration
- Reactive Configuration
- Health checks
How to dynamically **select archetypes**
Demo
05
Discussion
Recap

● We created DYNAMOS to abstract data-exchange scenarios with OS properties

● We generate *Microservice Chains* and assign *roles* to abstract archetypes

● We showed two execution architectures, and let the data stewards choose how jobs are executed

● Archetypes and architectures, can be dynamically changed by altering the knowledge base

● We developed a custom method to transfer data within ephemeral jobs
Do we adhere to these **OS properties**?

**Transparency**
Archetype and microservice architecture is transparent (invisible) to Ana

**Concurrency**
Data stewards handle requests concurrently

**Resource Management**
Change archetypes to re-distribute system load

**Scalability**
Every part of the system can be scaled individually

**Flexibility**
Optimum archetypes or architectures can be selected dynamically

**Security**
Built-in authorization flow, isolated jobs
Discussion

Latency is not that important

The microservices, sidecar pattern, and cold starts of ephemeral jobs add to latency

Flexibility and customizability are key

Dynamic archetypes, and architectures

Extendable algorithms for extra-functional properties
Threats to **validity**

- All taken measurements, where not statistically relevant
- Ignored HTTP as a communication method
Conclusion
Conclusion

Abstraction

Complexity of data-exchanges abstracted, adding data-exchange microservices is ‘easy’

Dynamic

Dynamical archetypes, architectures and extendible algorithms

Practical

We hope that this practical application of policy, archetypes, and microservices inspires future work toward an actual Digital Data-sharing Marketplace.
Future research

Data pods and authorization
Realize the full authorization flow towards a dataset in a data pod

Analyze more archetypes and scenarios
Does the microservice chain architecture hold for all scenarios?
Thanks!

Do you have any questions?

j.stutterheim@uva.nl

https://github.com/Jorrit05/DYNAMOS
Drinks in the ‘Oerknal’ afterwards

CREDITS: This presentation template was created by Slidesgo, and includes icons by Flaticon and infographics & images by Freepik

Credits:

Supervisor
Ana Oprescu

2nd reader
Thomas van Binsbergen

Review
Aleandro Mifsud
Marco Brohet
Jurriën Stutterheim

Frontend
Bart van der Hoeven

Discussion
Yuri Demchenko
Contributions

Research contributions

1. Combining data-sharing requests and DDM archetypes into microservice chains
2. Specific microservice requirements for generic microservices to perform the AMDEX UNL use case scenarios
3. Analysis of how baseline extra-functional properties (costs, energy, etc.) can be taken into account in the system
4. Analysis of different approaches to developing a distributed system for data-sharing purposes
5. Architecture to bridge the gap from a formal policy evaluation result, to a web environment where jobs are executed

Technical contributions

1. Generic method to create single-use, self-cleaning microservice jobs
2. Generic communication method of transferring data through microservice chains
3. Exact types, code, and templates to generate microservice chains
4. Loosely messaging solution, with exact message definitions for communication in DYNAMOS
5. An open-source system that allows experimentation with microservice chains in a distributed environment
<table>
<thead>
<tr>
<th>Key path</th>
<th>Contains</th>
</tr>
</thead>
<tbody>
<tr>
<td>/agents/ &lt; Agentname &gt;</td>
<td>Agent configuration</td>
</tr>
<tr>
<td>/agents/jobs/ &lt; Agent name &gt; / &lt; User name &gt; / &lt; Job name &gt;</td>
<td>Composition request data</td>
</tr>
<tr>
<td>/agents/jobs/ &lt; Agent name &gt; /queueInfo/ &lt; Job name &gt;</td>
<td>Queue attached to a job</td>
</tr>
<tr>
<td>/archetypes/ &lt; Archetype name &gt;</td>
<td>Archetype configuration</td>
</tr>
<tr>
<td>/datasets/ &lt; dataset name &gt;</td>
<td>Dataset metadata</td>
</tr>
<tr>
<td>/microservices/ &lt; Microservice name &gt; /chainMetadata</td>
<td>Microservice metadata</td>
</tr>
<tr>
<td>/policyEnforcer/agreements/ &lt; data steward name &gt;</td>
<td>Agreements for data-sharing</td>
</tr>
<tr>
<td>/requestTypes/ &lt; request type name &gt;</td>
<td>Request type information</td>
</tr>
</tbody>
</table>

Table 4.3: Etcd keys
// The sidecar definition.

service Microservice {
  rpc SendData(MicroserviceCommunication) returns (google.protobuf.Empty) {} 
}

message MicroserviceCommunication {
  string type = 1;
  string request_type = 2;
  google.protobuf.Struct data = 3;
  map<string, string> metadata = 4;
  google.protobuf.Any original_request = 5;
  RequestMetadata request_metadata = 6;
  map<string, bytes> traces = 7; // Binary or textual representation of span context
  bytes result = 8;
  repeated string routing_data = 9; // To be used for persistent jobs
}
Venus has a beautiful name and is the second planet from the Sun. It’s terribly hot, even hotter than Mercury, and its atmosphere is extremely poisonous.

**Current situation**

**Problems**

01. Wastefulness in production processes
02. Climate crisis
03. Making Water clean
Hypotheses

**Hypothesis A**
Mercury is the closest planet to the Sun and the smallest one in the Solar System—it’s only a bit larger than the Moon

**Hypothesis B**
Venus has a beautiful name and is the second planet from the Sun. It’s terribly hot—even hotter than Mercury—and its atmosphere is extremely poisonous

**Hypothesis C**
Jupiter is a gas giant and the biggest planet in the Solar System. It's the fourth-brightest object in the night sky
Study objectives

Methods for improving operations
Mercury is the closest planet to the Sun and the smallest of them all

Developing ways of reducing costs
Venus has a beautiful name and is the second planet from the Sun

Improving efficiency of processes
Despite being red, Mars is actually a cold place. It’s full of iron oxide dust
Mercury is the closest planet to the Sun

Mars is full of iron oxide dust, which gives the planet its reddish cast

Jupiter is a gas giant and the biggest planet in the Solar System

Venus has a beautiful name and is the second planet from the Sun

Earth is the third planet from the Sun and harbors life
Theoretical framework

Key terms
- Mercury is small
- Earth harbors life
- Jupiter is quite big

Relevant theories
- **Theory A**
  - Mercury is the smallest planet of them all
- **Theory B**
  - Venus is the second planet from the Sun

Our framework
- Venus has a beautiful name and is the second planet from the Sun. It’s hot and has a poisonous atmosphere
# Schedule

<table>
<thead>
<tr>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>Jun</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task 1</strong></td>
<td><strong>Task 2</strong></td>
<td><strong>Task 3</strong></td>
<td><strong>Task 1</strong></td>
<td><strong>Task 2</strong></td>
<td><strong>Task 3</strong></td>
</tr>
<tr>
<td>Neptune is far away from Earth</td>
<td>Mercury is a very small planet</td>
<td>Saturn is a gas giant</td>
<td>Despite being red, Mars is cold</td>
<td>Earth is the planet with life</td>
<td>Venus has a beautiful name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Jan 1 - Mar 15</td>
<td>Feb 1 - Apr 30</td>
<td>May 15 - Jun 30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>In progress</td>
<td>Delayed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Completed</td>
<td></td>
</tr>
</tbody>
</table>
Methodology

Type of data
Earth is the third planet from the Sun and the only one that harbors life in the Solar System

Motives
Venus has a beautiful name and is the second planet from the Sun. It’s terribly hot—even hotter than Mercury

Data collection
Despite being red, Mars is actually a cold place. It’s full of iron oxide dust, which gives the planet its reddish cast

Specific sampling
Saturn is a gas giant and has several rings. It’s composed mostly of hydrogen and helium
Phase A

• Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's a bit larger than the Moon

• Jupiter is a gas giant, the biggest planet in the Solar System and the fourth-brightest object in the night sky

• Neptune is the farthest planet from the Sun. It's also the fourth-largest planet by diameter in the Solar System

Phase B

• Venus has a beautiful name and is the second planet from the Sun. It’s terribly hot—even hotter than Mercury

• Saturn is a gas giant and has several rings. This planet is composed mostly of hydrogen and helium

• Earth is the third planet from the Sun and the only one that harbors life in the Solar System. This is where we all live
Analysis & development

Mercury is the closest planet to the Sun and the smallest one in the Solar System. The planet’s name has nothing to do with the liquid metal.

- Venus has a beautiful name and is the second planet from the Sun.
- Earth is the third planet from the Sun and the only one that harbors life in the Solar System.
- Despite being red, Mars is cold.

Venus
- Venus is very hot.

Mercury
- Mercury is small.

Follow the link in the graph to modify its data and then paste the new one here. For more info, click here.
Analysis of the **results**

- **Venus**: Venus is very hot
- **Mercury**: Mercury is small
- **Mars**: Despite being red, Mars is actually cold

Follow the link in the graph to modify its data and then paste the new one here. For more info, click here
Top industrial engineering programs

Stanford University **US**

University of Cambridge **UK**

ETH Zurich **Switzerland**

Tsinghua University **CHN**
Discussion

Despite being red, Mars is actually a cold place. It's full of iron oxide dust, which gives the planet its reddish cast. Earth is the third planet from the Sun and the only one that harbors life in the Solar System. This is where we all live:

- Ceres is located in the main asteroid belt
- The Moon is Earth’s natural satellite
- Neptune is very far away from us
- Pluto is now considered a dwarf planet
Some **conclusions**

**System optimization**
Despite being red, Mars is actually a cold place.

**Broad application**
Mercury is the smallest planet of them all.

**Interdisciplinary approach**
Venus is extremely hot, even more than Mercury.

**Human-centered focus**
Saturn is a gas giant and has several rings.
Bibliographical references

Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Surname, A. (YEAR). *Name of the source*. Publisher
Mockups

You can replace the images on the screen with your own work. Just right-click on them and select “Replace images”
Alternative **resources**

Here’s an assortment of alternative resources whose style fits the one of this template:

- Technology background in abstract style
Resources

Did you like the resources in this template? Get them for free at our other websites:

Photos

- Close up on person working on alternative energy
- Medium shot woman working as engineer

Vector

- Technology background in abstract style
Instructions for use

If you have a free account, in order to use this template, you must credit Slidesgo by keeping the Thanks slide. Please refer to the next slide to read the instructions for premium users.

As a Free user, you are allowed to:
- Modify this template.
- Use it for both personal and commercial projects.

You are not allowed to:
- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit our blog: https://slidesgo.com/faqs and https://slidesgo.com/slidesgo-school
Instructions for use (premium users)

As a Premium user, you can use this template without attributing Slidesgo or keeping the "Thanks" slide.

You are allowed to:
- Modify this template.
- Use it for both personal and commercial purposes.
- Hide or delete the “Thanks” slide and the mention to Slidesgo in the credits.
- Share this template in an editable format with people who are not part of your team.

You are not allowed to:
- Sublicense, sell or rent this Slidesgo Template (or a modified version of this Slidesgo Template).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit our blog: https://slidesgo.com/faqs and https://slidesgo.com/slidesgo-school
This presentation has been made using the following fonts:

Kantumruy Pro
(https://fonts.google.com/specimen/Kantumruy+Pro)

#263165 #f4eef4 #f3f4ee #ffffff
Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out how it works.
You can easily resize these resources without losing quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Group the resource again when you're done. You can also look for more infographics on Slidesgo.
...and our sets of editable icons

You can resize these icons without losing quality.
You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.
In Google Slides, you can also use Flaticon’s extension, allowing you to customize and add even more icons.
Nature Icons
SEO & Marketing Icons
### Contents of this template

You can delete this slide when you’re done editing the presentation

<table>
<thead>
<tr>
<th><strong>Fonts</strong></th>
<th>To view this template correctly in PowerPoint, download and install the fonts we used</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Used and alternative resources</strong></td>
<td>An assortment of graphic resources that are suitable for use in this presentation</td>
</tr>
<tr>
<td><strong>Thanks slide</strong></td>
<td>You must keep it so that proper credits for our design are given</td>
</tr>
<tr>
<td><strong>Colors</strong></td>
<td>All the colors used in this presentation</td>
</tr>
<tr>
<td><strong>Icons and infographic resources</strong></td>
<td>These can be used in the template, and their size and color can be edited</td>
</tr>
<tr>
<td><strong>Editable presentation theme</strong></td>
<td>You can edit the master slides easily. For more info, click <a href="#">here</a></td>
</tr>
</tbody>
</table>

For more info: [SLIDESGO](#) | [BLOG](#) | [FAQs](#)  
You can visit our sister projects: [FREEPIK](#) | [FLATICON](#) | [STORYSET](#) | [WEPIK](#) | [VIDEVO](#)