



# ANTI-AGING BEHANDLUNG FÜR LEGACY SOFTWARE

JÖRG BÄCHTIGER



# Disclaimer

These materials have been prepared by Finnova AG. These materials are based on information provided by or on behalf of the Company and/or other potential transaction participants, from public sources or otherwise reviewed by Finnova AG. Finnova AG assumes no responsibility for the presented information. These materials do not constitute an offer or solicitation to sell or purchase any products and are not a commitment by Finnova AG to provide or arrange it. Finnova AG assumes no obligation to update or otherwise revise these materials. These materials have not been prepared with a view toward public disclosure under state or federal laws or otherwise, are intended for the benefit and use of the Company, and may not be reproduced, disseminated, quoted, summarized or referred to, in whole or in part, without the prior written consent of Finnova AG. These materials may not reflect information known to other professionals in other business areas of Finnova AG. Finnova AG is a full service firm providing banking and other services and products to a wide range of corporations, domestically and offshore, from which conflicting interests or duties may arise. In the ordinary course of these activities, Finnova AG may at any time hold long or short positions. Any information given in this presentation does not represent any binding obligation to provide any product or service to anyone. Nor can a customer or potential customer derive any product or product properties from the presented information. Finnova has the right to withdraw any covenants which are made in the context of this presentation without any consequences so ever. Furthermore all rights for the presentation reserved with the exclusion of content, which is a copyright protected.

# Content

Field report

Works for Finnova

# **Finnova AG**

Banking software

Since 1974

More than 100 banks

# About me

Sinclair ZX-81

Scala, Akka

# Anti-Aging?

1500 users

10MLOC

1500 man years

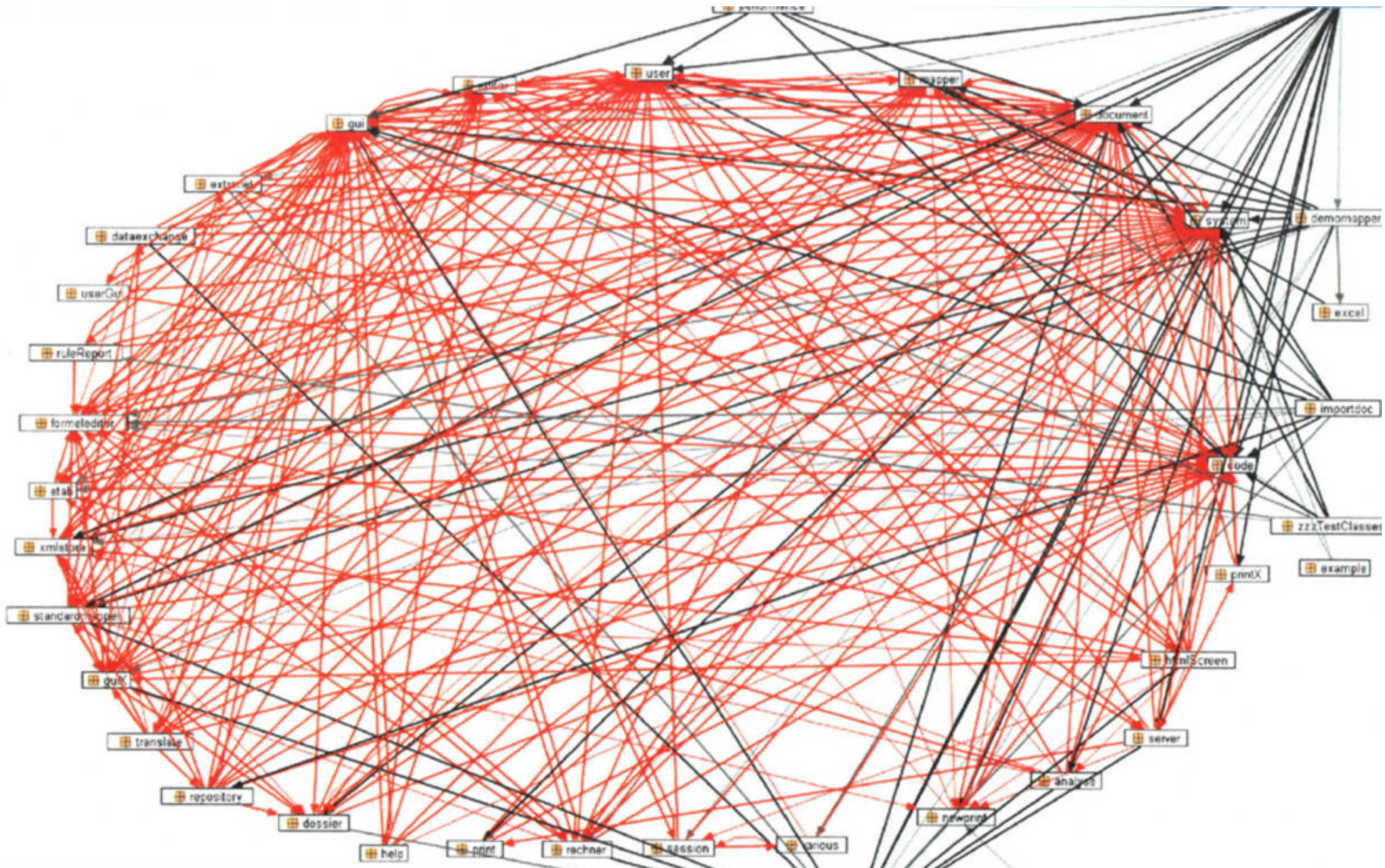
PL/SQL, Java, C#, DB

Automated tests

Single source base



# Technical-Debt





# Goal

Modular, Composable

Scalable

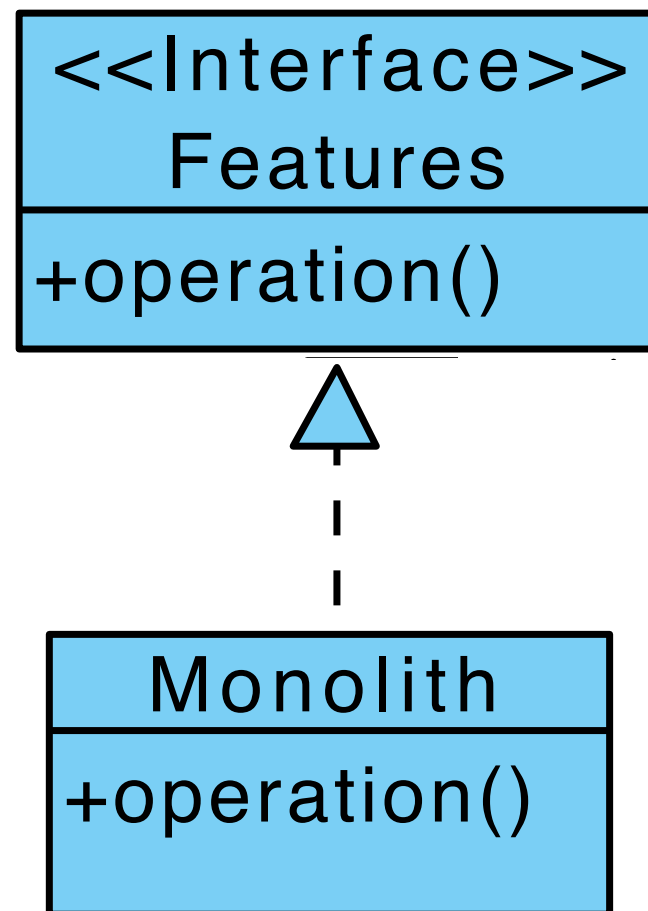
Process-oriented

Technology independent

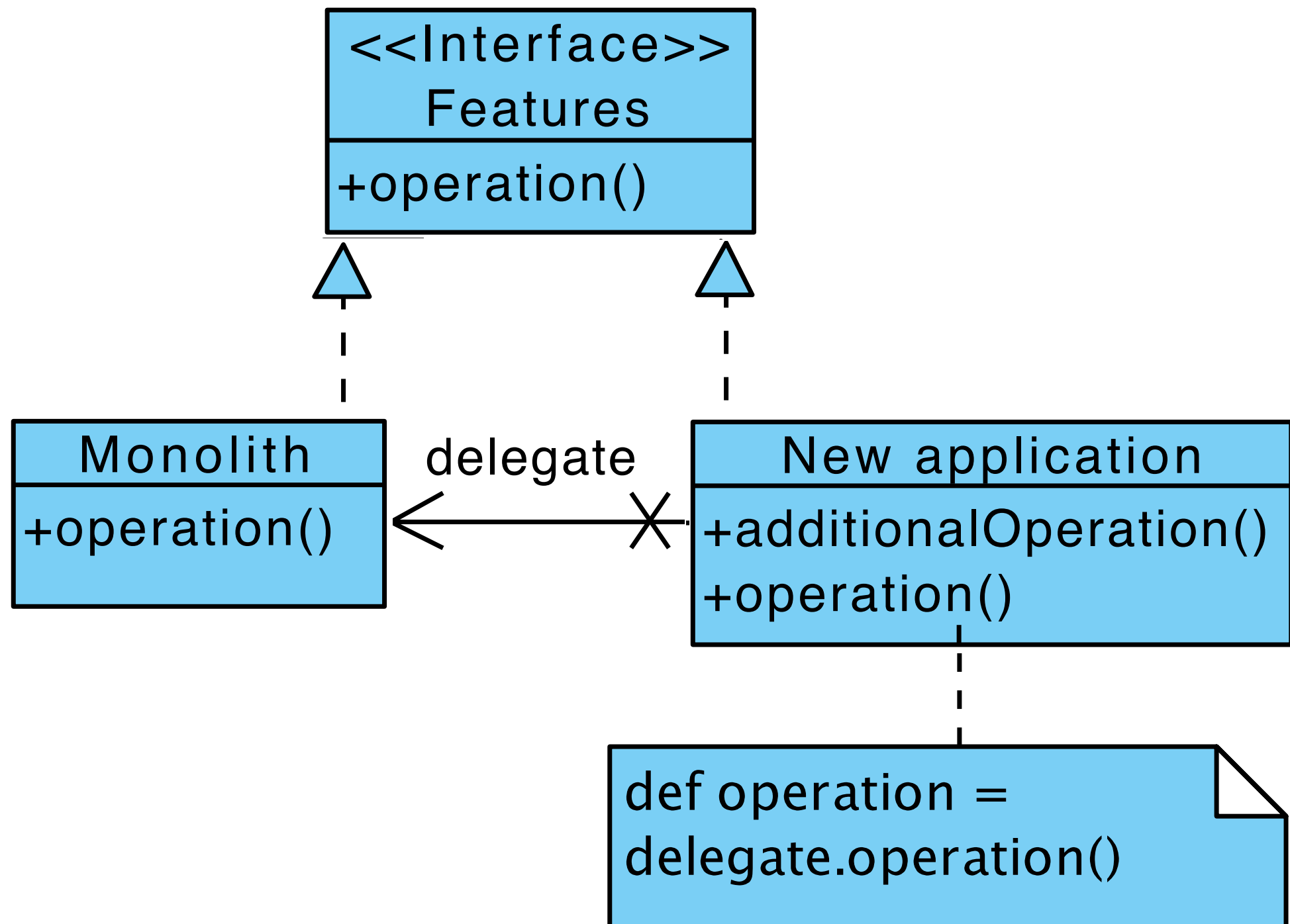
Easy maintainable/develop



# Concept



# Concept



# Concept

# uServices

Modular, Composable

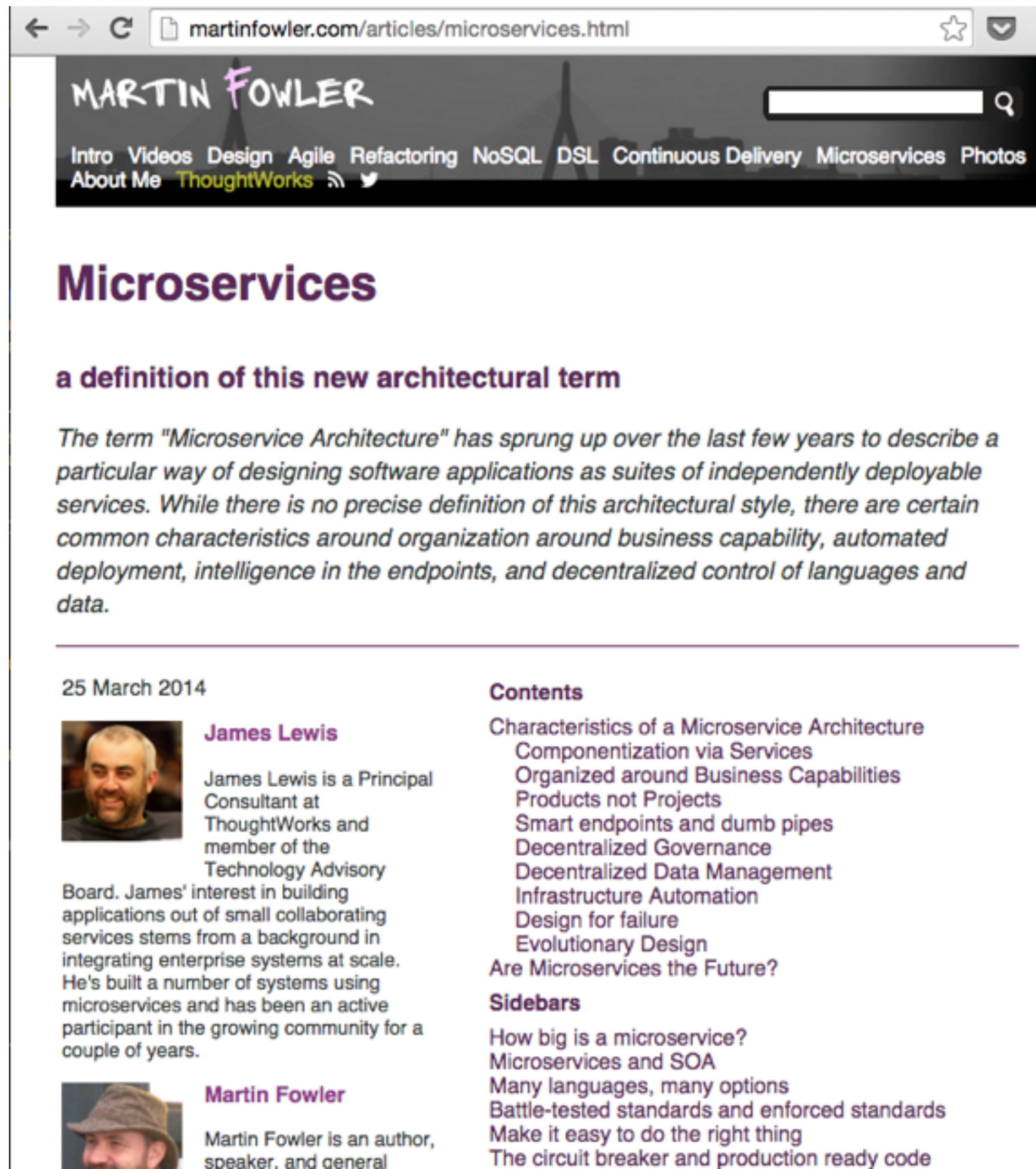
Scalable

Process-oriented

Technology independent

Easy maintainable/develop

# Concept uServices



The screenshot shows a web browser window with the address bar displaying 'martinfowler.com/articles/microservices.html'. The page header features the name 'MARTIN FOWLER' in a stylized font, a search bar, and a navigation menu with links: 'Intro', 'Videos', 'Design', 'Agile', 'Refactoring', 'NoSQL', 'DSL', 'Continuous Delivery', 'Microservices', and 'Photos'. Below the header, the article title 'Microservices' is prominently displayed in a large, bold, purple font. Underneath the title is a subtitle 'a definition of this new architectural term' in a smaller, bold, purple font. The main body of the article begins with a paragraph explaining the term 'Microservice Architecture'. Below this paragraph, there is a horizontal line separating the main content from a sidebar. The sidebar is divided into two columns. The left column contains a date '25 March 2014', a profile picture of James Lewis, his name 'James Lewis', and a bio describing him as a Principal Consultant at ThoughtWorks. Below this is another profile picture of Martin Fowler, his name 'Martin Fowler', and a bio describing him as an author, speaker, and general. The right column of the sidebar is titled 'Contents' and lists the following topics: 'Characteristics of a Microservice Architecture', 'Componentization via Services', 'Organized around Business Capabilities', 'Products not Projects', 'Smart endpoints and dumb pipes', 'Decentralized Governance', 'Decentralized Data Management', 'Infrastructure Automation', 'Design for failure', 'Evolutionary Design', 'Are Microservices the Future?', 'Sidebars', 'How big is a microservice?', 'Microservices and SOA', 'Many languages, many options', 'Battle-tested standards and enforced standards', 'Make it easy to do the right thing', and 'The circuit breaker and production ready code'.

← → ↻ [martinfowler.com/articles/microservices.html](http://martinfowler.com/articles/microservices.html) ☆

MARTIN FOWLER

Intro Videos Design Agile Refactoring NoSQL DSL Continuous Delivery Microservices Photos  
About Me ThoughtWorks


## Microservices

a definition of this new architectural term

*The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.*


---

25 March 2014



**James Lewis**

James Lewis is a Principal Consultant at ThoughtWorks and member of the Technology Advisory Board. James' interest in building applications out of small collaborating services stems from a background in integrating enterprise systems at scale. He's built a number of systems using microservices and has been an active participant in the growing community for a couple of years.



**Martin Fowler**

Martin Fowler is an author, speaker, and general

### Contents

- Characteristics of a Microservice Architecture
  - Componentization via Services
  - Organized around Business Capabilities
  - Products not Projects
  - Smart endpoints and dumb pipes
  - Decentralized Governance
  - Decentralized Data Management
  - Infrastructure Automation
  - Design for failure
  - Evolutionary Design
- Are Microservices the Future?

### Sidebars

- How big is a microservice?
- Microservices and SOA
- Many languages, many options
- Battle-tested standards and enforced standards
- Make it easy to do the right thing
- The circuit breaker and production ready code



# Concept

# uServices

In short, the microservice **architectural style** [1] is an approach to developing a single application as a **suite of small services**, each running in its **own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and **independently deployable** by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

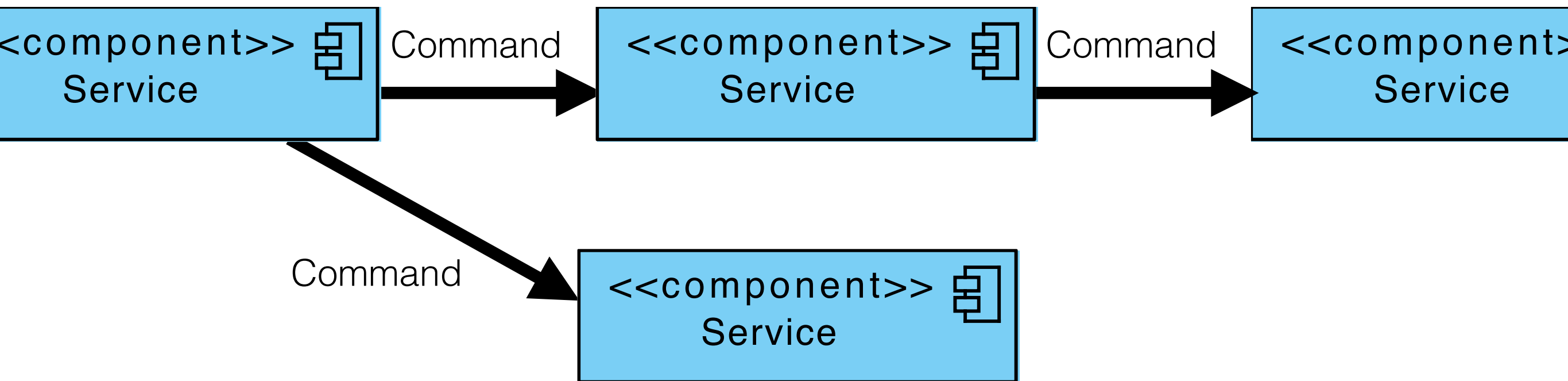
# Concept

# uServices

build for replacement  
not for reuse

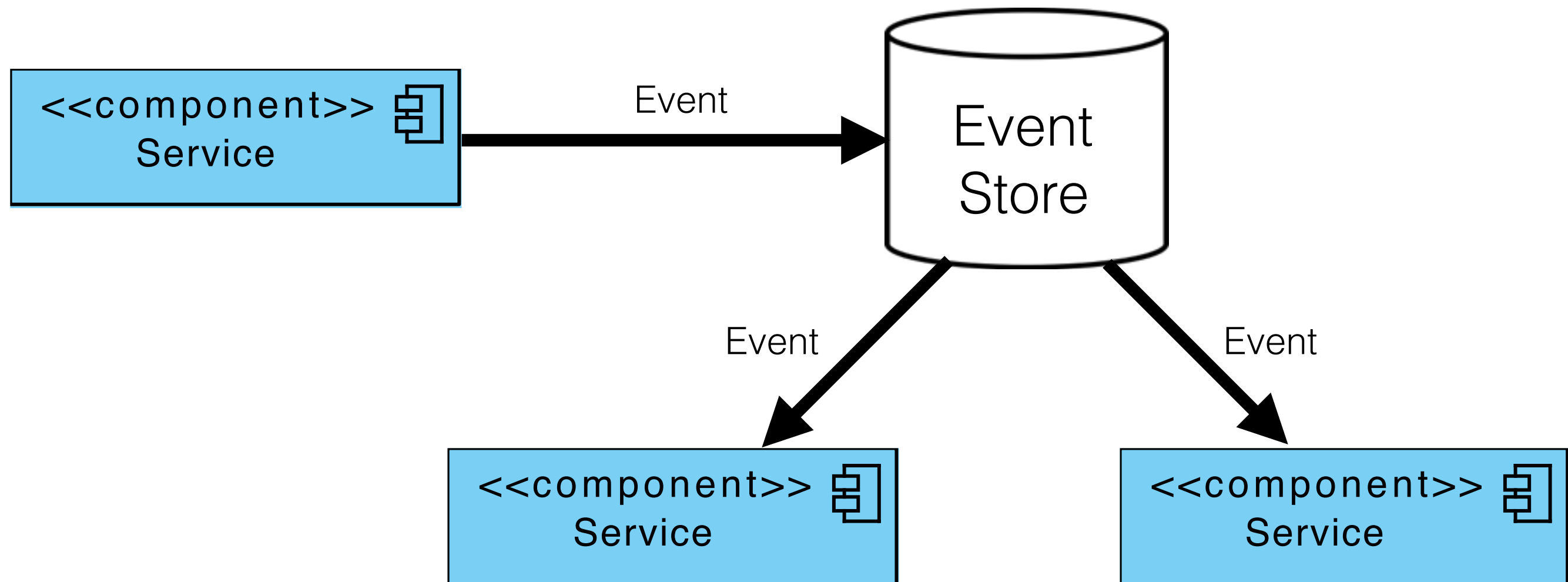
# uServices

Modular, Composable



# uServices

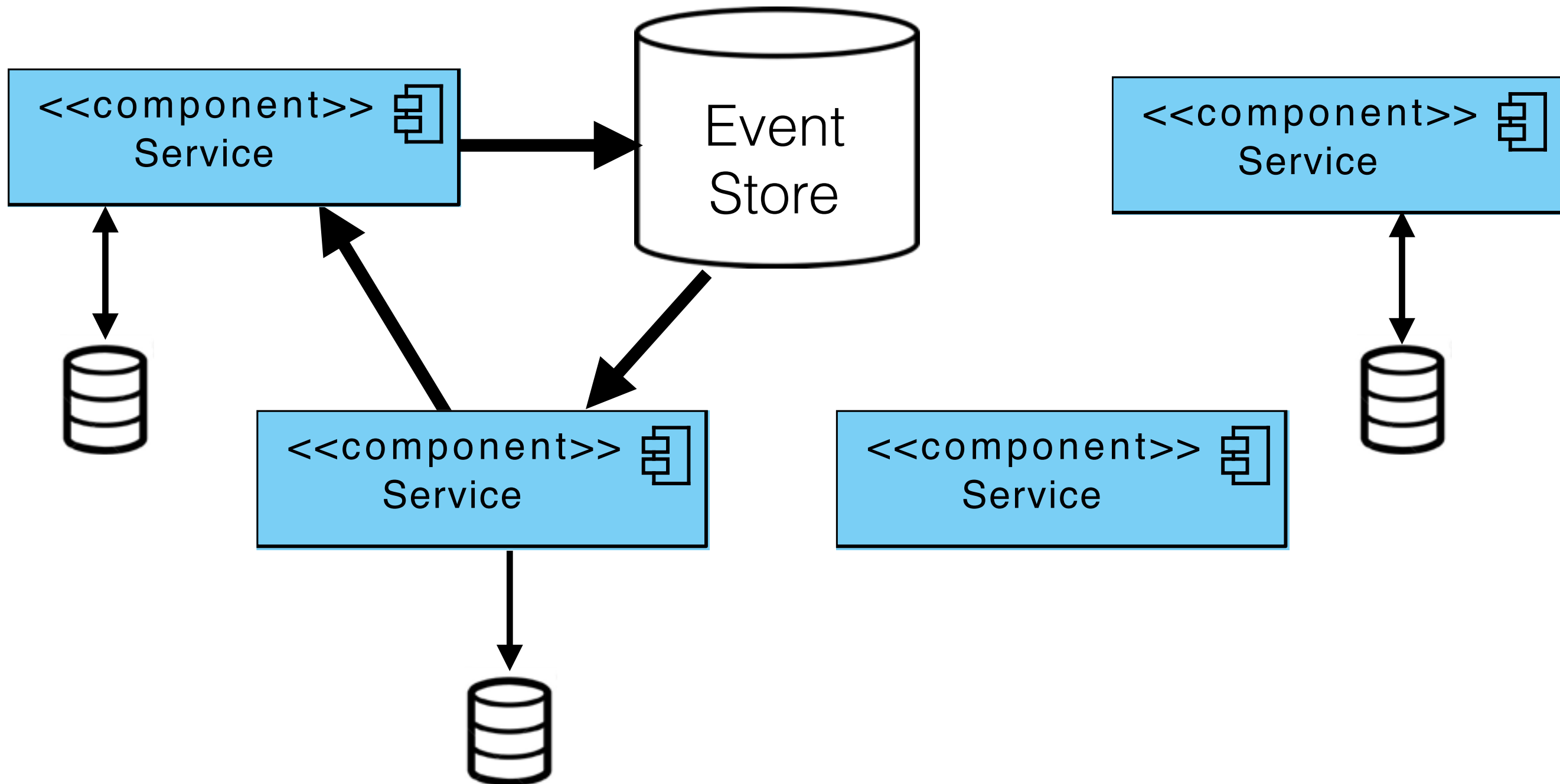
Modular, Composable





# uServices

Modular, Composable



# uServices

Modular, Composable

The screenshot displays a modular banking application interface. On the left, a sidebar contains navigation icons for user management, account management, and product selection. The main content area is titled 'Produktauswahl' (Product Selection) and features a search bar with filters for 'Konto' (Account) and 'Karte' (Card). Below the search bar, a list of products is shown, including 'Privatkonto' (Private Account), 'Sparkonto' (Savings Account), 'Privatkonto 60plus', 'Maestrokarte' (MasterCard), 'Visa-Karte', 'Mastercard', 'American Express', and 'Diners Club'. Each product entry includes its name, ID, and availability status. The right panel shows the 'Produkteübersicht' (Product Overview) and 'PRODUKTDDETAIL' (Product Details) for a selected 'Sparkonto'. The details section includes fields for 'Arbeitsgebiet' (Work Area), 'Sperrcode' (Lock Code), 'Sitz' (Seat), 'Buchungsverdichtung' (Booking Density), 'Kontokorrent' (Current Account), 'Saldo unterdrücken' (Suppress Balance), 'Zinsausanz' (Interest Period), 'Konto-Bonität' (Account Creditworthiness), 'Währung' (Currency), 'Spesenfrei' (Fee-Free), and 'Monatsauszug' (Monthly Statement). A 'VERWERFEN' (Reject) button and a 'WEITER' (Next) button are at the bottom right.

<<component>>  
Service



<<component>>  
Service



<<component>>  
Service



# uServices

# Polymer

<!--

*Element to edit a list of email addresses.*

##### *Example*

```
<fin-emails types="{{ types }}" data="{{ data }}" errors="{{ errors }}"></fin-emails>
```

*@element fin-emails*

*@blurb Element to edit a list of email addresses.*

*@status alpha*

*@demo demo.html*

-->

```
<dom-module id="fin-emails">
```

```
  <style>
```

```
    :host {  
      display: block;  
    }
```

```
    paper-checkbox {  
      padding: 10px 10px 10px 0;  
    }
```

```
    .email-type,  
    .email-address,  
    .remarks {  
      margin-bottom: 1em;  
    }
```

```
    paper-icon-button[icon="home"] {  
      color: lightgray;  
    }
```

# uServices

# Polymer

```
<template>
  <fin-card table layout invalid$="[[ _computeInvalidity(errors) ]]">
    <header class="fin-card-header"><h3>[[ _label(labels, 'header', 'E-Mails') ]]</h3></header>
    <article hidden$="[[ _hasNoItems(data.*)]]">
      <template is="dom-repeat" items="{{ data }}" as="email">

        <header class="toolbar" invalid$="[[ _computeRecordInvalidity(errors, index) ]]"
          main-address$="[[ email.isMainAddress ]]">
          <div col><strong>{{ email.emailAddress }}</strong></div>
          <div>
            <paper-icon-button icon="home" on-tap="_setMainAddress"></paper-icon-button>
            <paper-icon-button icon="delete" on-tap="_removeEmailAction"></paper-icon-button>
            <paper-icon-button icon="create" on-tap="_toggleCollapsed"></paper-icon-button>
          </div>
        </header>

        <article hidden$="[[ _isCollapsed(_collapsed, index) ]]" invalid$="[[ invalid ]]">
          <div class="email-type">
            <fin-dropdown
              label="[[ _label(labels, 'types', 'Type') ]]"
              options="[[ types ]]"
              invalid="[[ _findError(errors, 'typeOfEMail', index) ]]"
              error-message="[[ _findError(errors, 'typeOfEMail', index) ]]"
              selected="{{ { email.typeOfEMail } }}"></fin-dropdown>
          </div>

          <div class="email-address">
            <fin-input
              label="[[ _label(labels, 'email', 'E-Mail') ]]"
              invalid="[[ _findError(errors, 'emailAddress', index) ]]"
```



# uServices

# Polymer

```
properties: {  
  /*  
   * the selectable email types  
   *  
   * [ { id: 1, name: "E-type 1" }, { id: 2, name: "E-type 2" }, { id: 3, name: "E-type 3" } ]  
   */  
  types: {  
    type: Array,  
    value: []  
  },  
  
  /*  
   * the whole set of all data records  
   *  
   * [{  
   *   "isMainAddress": true, # Boolean  
   *   "newsletterAllowed": true, # Boolean  
   *   "typeOfEMail": 2, # id as reference to types  
   *   "emailAddress": "a@example.com", # String  
   *   "remarks": "remark 1" # String  
   * }]  
   */  
  data: {  
    type: Array,  
    value: [],  
    notify: true  
  },  
  
  /*  
   * Error messages to show. The example will show an error
```

# uServices

# Polymer

```
_hasNoItems: function (record) {  
  return !record.base || !record.base.length;  
},  
  
_isCollapsed: function (collapsed, index) {  
  if (collapsed && typeof(collapsed[index]) !== 'undefined') {  
    return collapsed[index];  
  } else {  
    return true;  
  }  
},  
  
_toggleCollapsed: function (event) {  
  this._toggleCollapsedAt(event.model.index);  
},  
  
_toggleCollapsedAt: function (index) {  
  var arr = this.get('_collapsed');  
  arr[index] = !this._isCollapsed(this.get('_collapsed'), index);  
  this.set('_collapsed', arr.slice());  
},  
  
_label: function (labels, key, defaultLabel) {  
  return labels[key] || defaultLabel;  
},  
  
_setMainAddress: function (event) {  
  for (var i in this.get('data')) {  
    this.set('data.' + i + 'isMainAddress', i == event.model.index)
```

# uServices

# Polymer

```
<link rel="import" href="http://assets.uniquedomain:14000/polymer/polymer.html">
<link rel="import" href="http://assets.uniquedomain:14000/paper-button/paper-button.html">
<link rel="import" href="http://assets.uniquedomain:14000/paper-material/paper-material.html">
<link rel="import" href="http://assets.uniquedomain:14000/fin-emails/fin-emails.html">
<!--
```

*Element providing form for the 3rd step of the Customer Onboarding Process.*

*##### Example*

```
    <cop-contact-context></cop-contact-context>
    @element cop-contact-context
    @status alpha
-->
<dom-module id="cop-context-contact">
  <link rel="import" type="css" href="../shared/shared.css">
  <template>
    <div class="cards">
      <fin-emails
        id="contactEmails"
        class="business-unit"
        types="[[metadata.mailTypes]]"
        data="{ {data.emailAddresses} }"
        errors="[[ _errors(errors, 'emailAddresses') ]]"
        labels="[[i18n(labels.emails)]]">
      </fin-emails>
```

# uServices

Modular, Composable

Scalable

Process-oriented

Technology independent

Easy maintainable/develop



# uServices

Technology independent

Easy maintainable/develop



DbC, SOC, SRP, ISP

# uServices

Technology independent

Easy maintainable/develop

`<<component>>`  
Service



# uServices

Technology independent

Easy maintainable/develop

`<<component>>`  
Service



$\mu$  ?

# uServices

Modular, Composable

Scalable

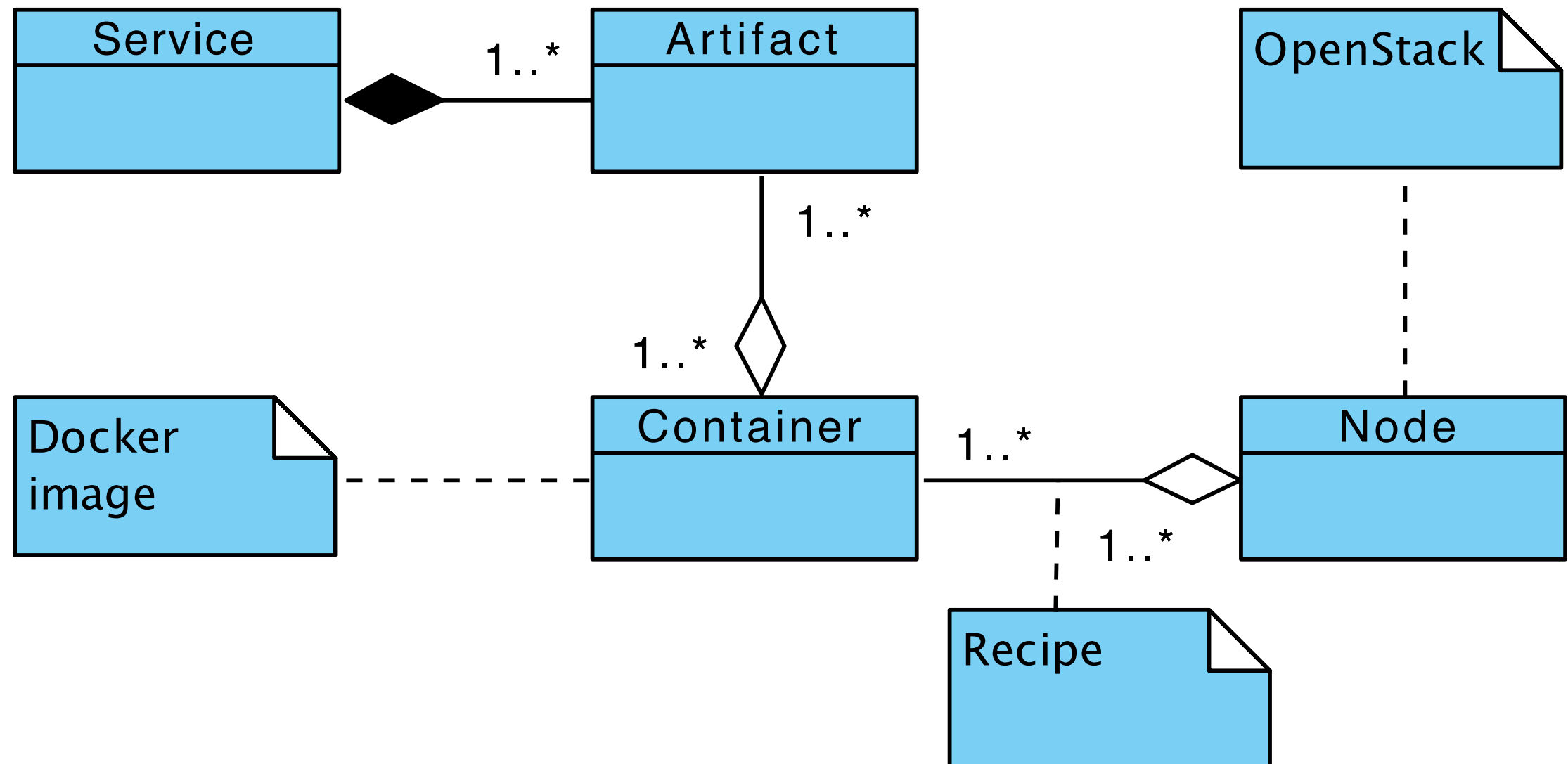
Process-oriented

Technology independent

Easy maintainable/develop

# uServices

Scalable



# uServices

## Scalable

heat\_template\_version: 2013-05-23

description: Example

parameters:

flavor:

type: string

description: The ID or name of the flavor to boot onto.

default: "m1.small"

external\_network:

type: string

description: UUID of an existing external network.

internal\_network:

type: string

description : UUID of an existing internal network.



# uServices

## Scalable

[Service]

User=core

Group=core

EnvironmentFile=/etc/environment

TimeoutStartSec=0

ExecStartPre=-/usr/bin/docker kill oidcdb

ExecStartPre=-/usr/bin/docker rm oidcdb

ExecStartPre=/usr/bin/docker pull dev.finnova.ch:8443/oic-mys

ExecStart=/usr/bin/docker run --name="oidcdb" -p 3306:3306

ExecStop=/usr/bin/docker stop oidcdb

- name: sauce-connect.service

command: start

content: |

[Unit]

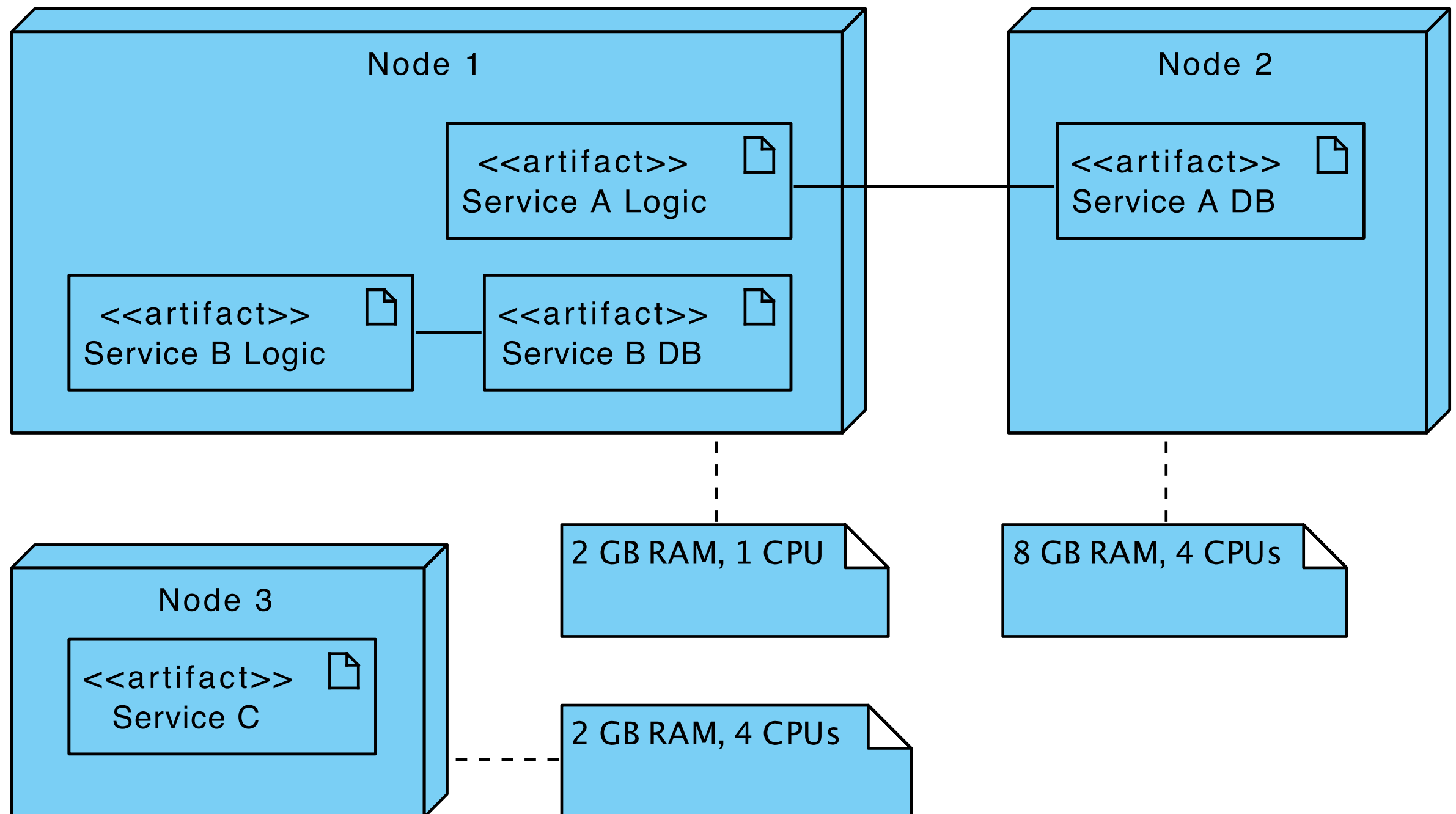
Description=Sauce Connect Tool

Requires=network-online.target

# uServices

Scalable

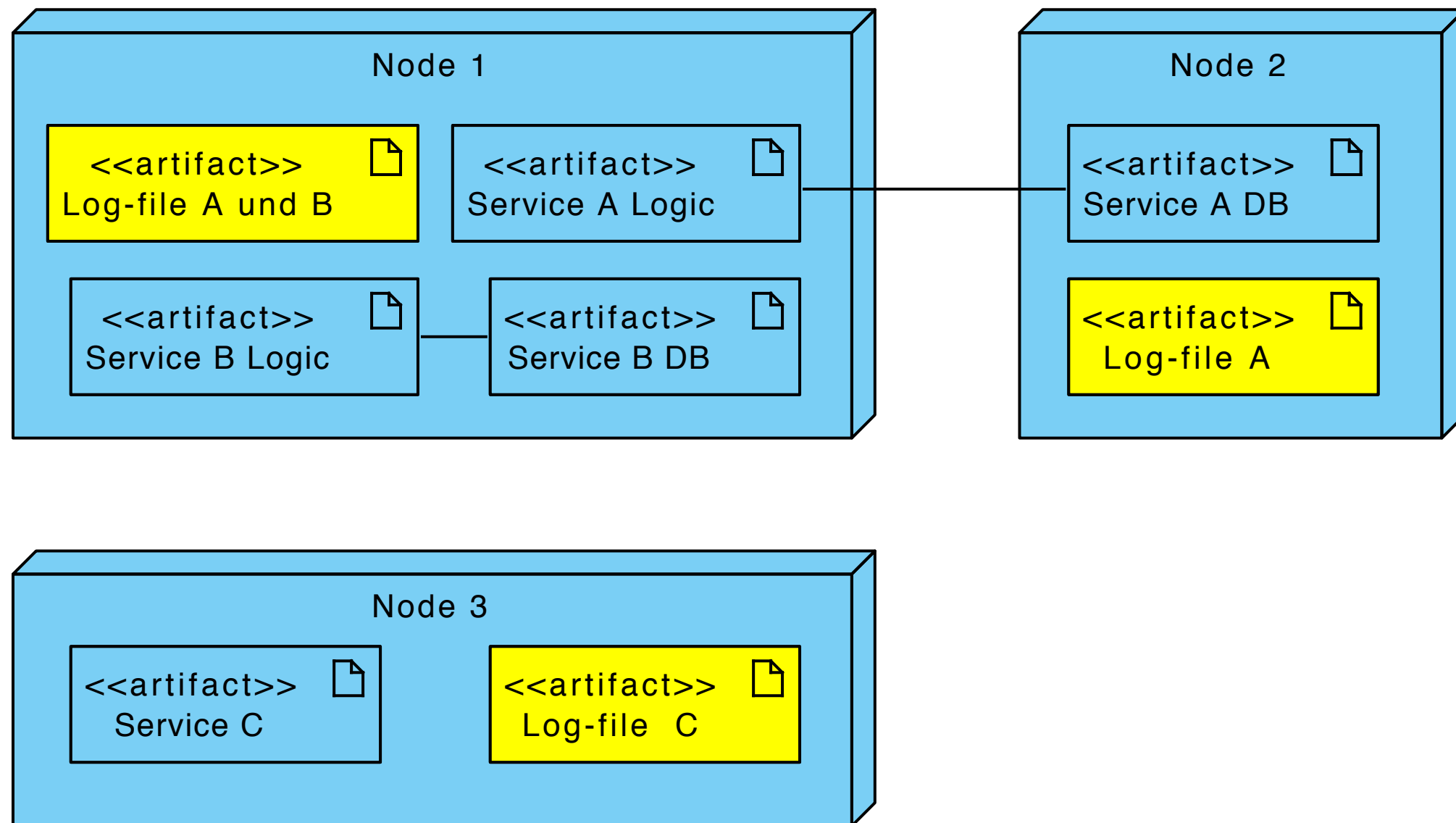
Monitoring



# uServices

Scalable

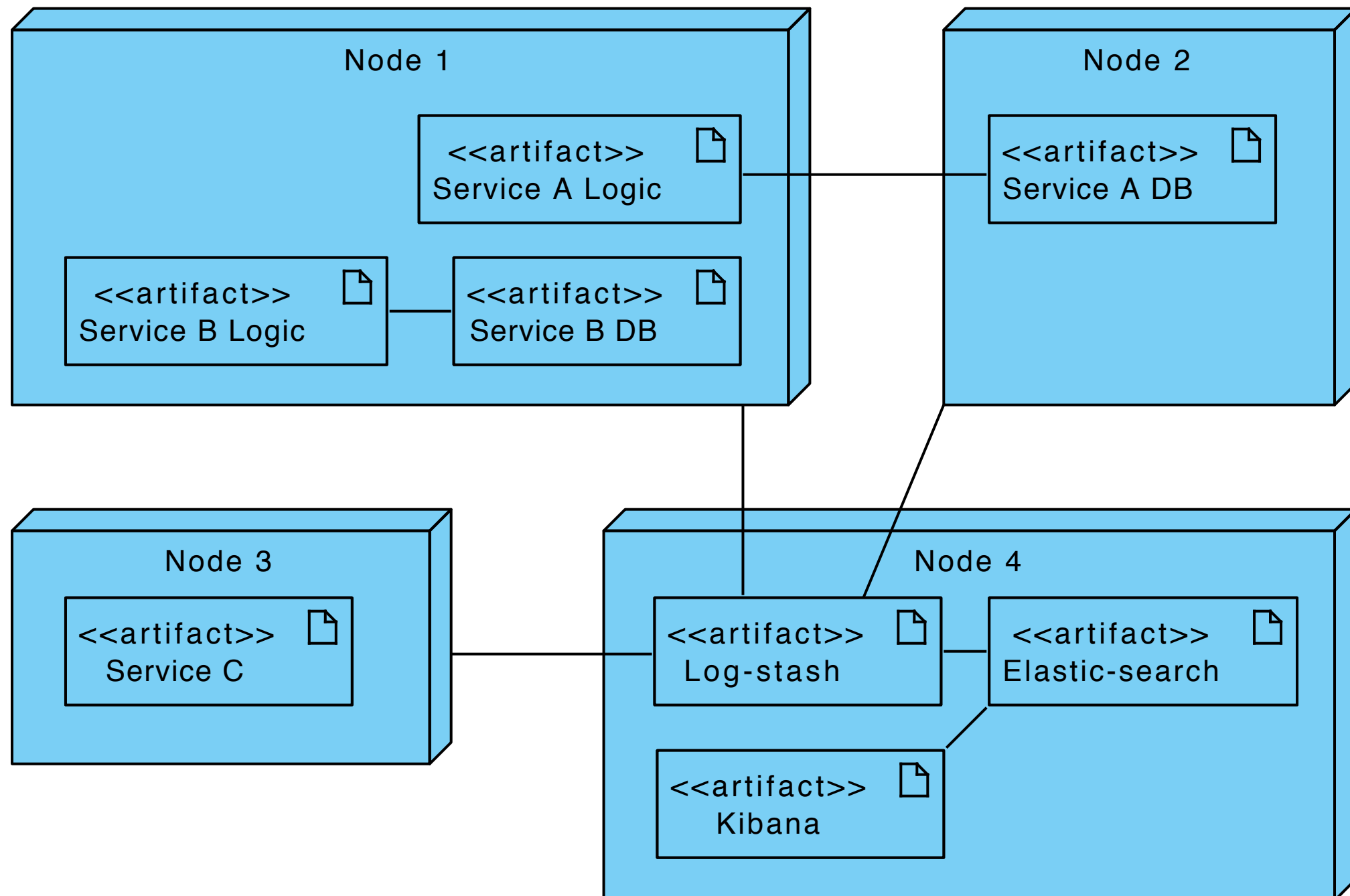
Monitoring



# uServices

Scalable

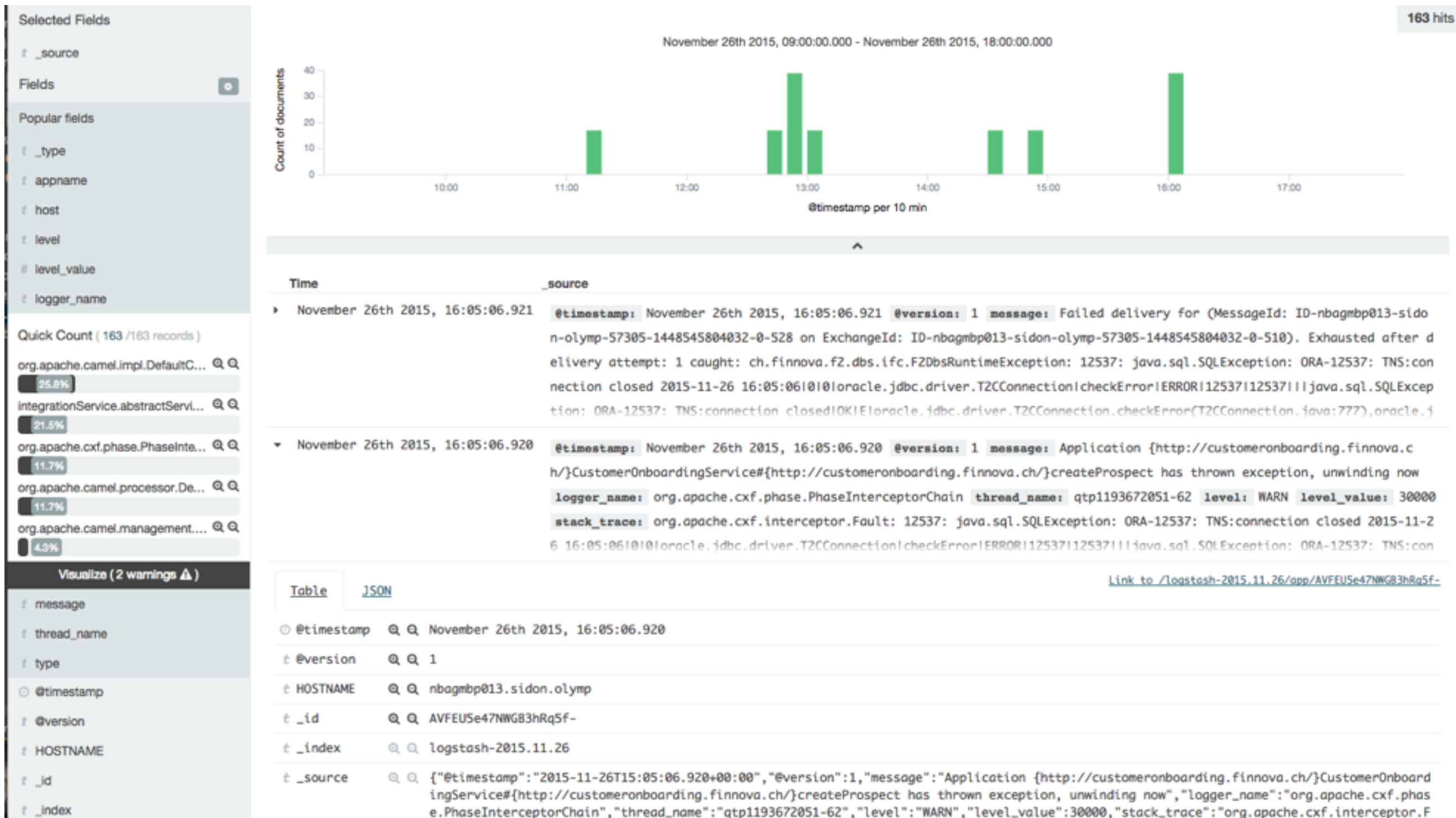
Monitoring



# uServices

## Scalable

## Monitoring



# Concept

Modular, Composable

Scalable

Process-oriented

Technology independent

Easy maintainable/develop



# Concept

## Process-oriented

The screenshot displays a CRM software interface with a top toolbar and a main data entry area. The main area is divided into several sections: a top header bar, a tabbed navigation bar, and three main content panels (Info, Person, and Person Zusatz).

**Top Header Bar:**

|          |         |          |             |                       |       |          |  |              |    |
|----------|---------|----------|-------------|-----------------------|-------|----------|--|--------------|----|
| Kunde... | 500.876 | Kurzname | Muster Hans | Status                | Aktiv | Sperrung |  | Domizil      | CH |
| Symbol   |         | Adresse  | Hans Muster | 4 Saisons, chemin des |       |          |  | 1967 Bramois |    |

**Tabbed Navigation Bar:**

|                            |                     |                 |                 |              |                |                      |                   |
|----------------------------|---------------------|-----------------|-----------------|--------------|----------------|----------------------|-------------------|
| Regulatorische Beziehungen | Pendenzen / Notizen | Alpha / Symbole | Zusatzfelder    | Auskünfte    | Finanz. Privat | Teilnehmer / Branche | Bilanzeinreichung |
| <b>Stamm</b>               | Adressen            | GeheimAdr       | Klassifizierung | Kundenprofil | GwV            | VSB                  | CRM Beziehungen   |

**Info Panel:**

Kundennummer \* 500876      Status \* 0 Aktiv      Eröffnungsdatum 05.08.2009

**Person Panel:**

Anrede \* Herr  
Titel  
Name \* Muster  
Name amtlich  
Vorname Hans  
Vorname amtlich  
Name vor Heirat  
Name vor Heirat amtlich  
☐ Ehemann: Name der Frau vorangestellt

**Person Zusatz Panel:**

Sprache \* Deutsch  
Geburtsdatum      Sterbedatum  
Land Geburtsort...      Geburtsort  
Nationalität... Schweiz  
Nationalität 2...  
Zivilstand \* Ledig      Zivilstand seit  
Güterstand <kein>  
Geschlecht Männlich  
Heimatorte...

**Person Zusatz II Panel:**

Briefanrede Sehr geehrter Herr Muster  
Ind. Briefanrede  
Bezeichnung lang Herr Hans Muster, Bramois ☒  
AHV-Nr./Soz.-Nr.  
Beruf  
Konfession  
Aufenth.-Bew.      bis

**Betreuer Panel:**

| Rolle              | Betreuer   |
|--------------------|--|
| 5000 Hauptbetreuer | Crufer Mauro <input checked="" type="checkbox"/> |

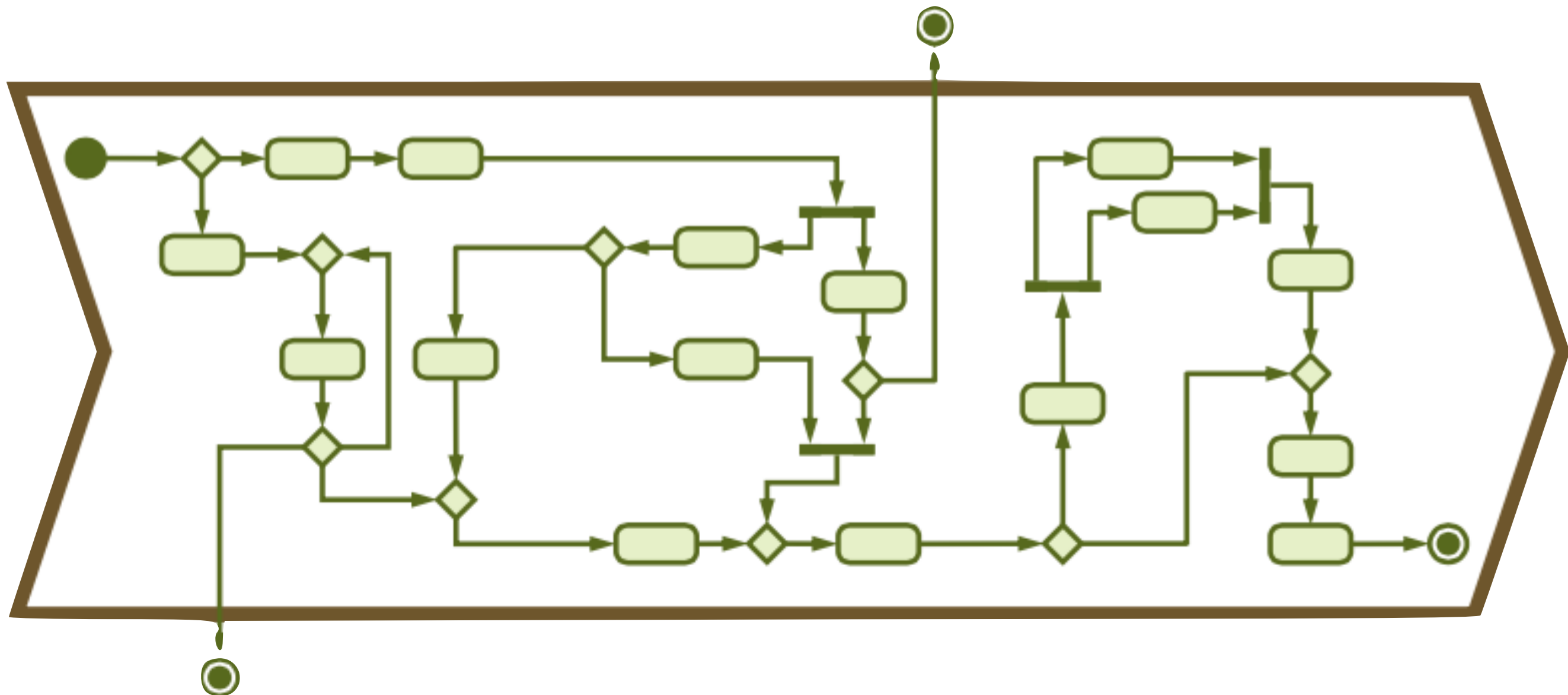
# Concept

Process-oriented



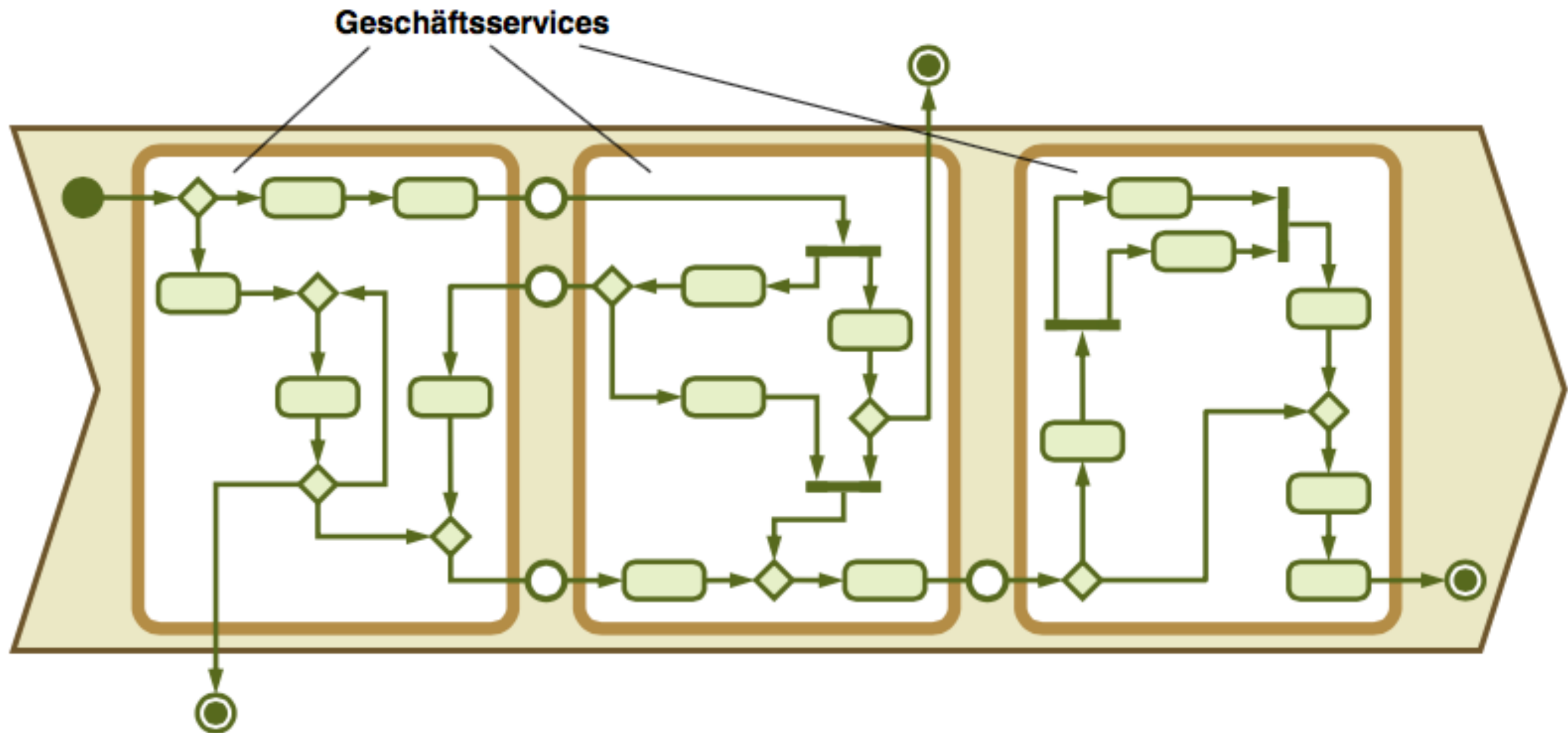
# Concept

Process-oriented



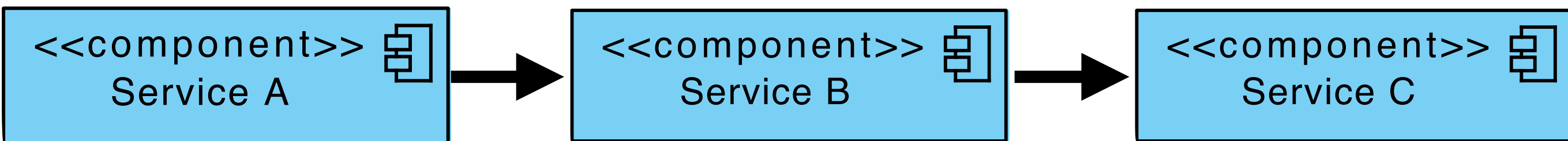
# Concept

## Process-oriented

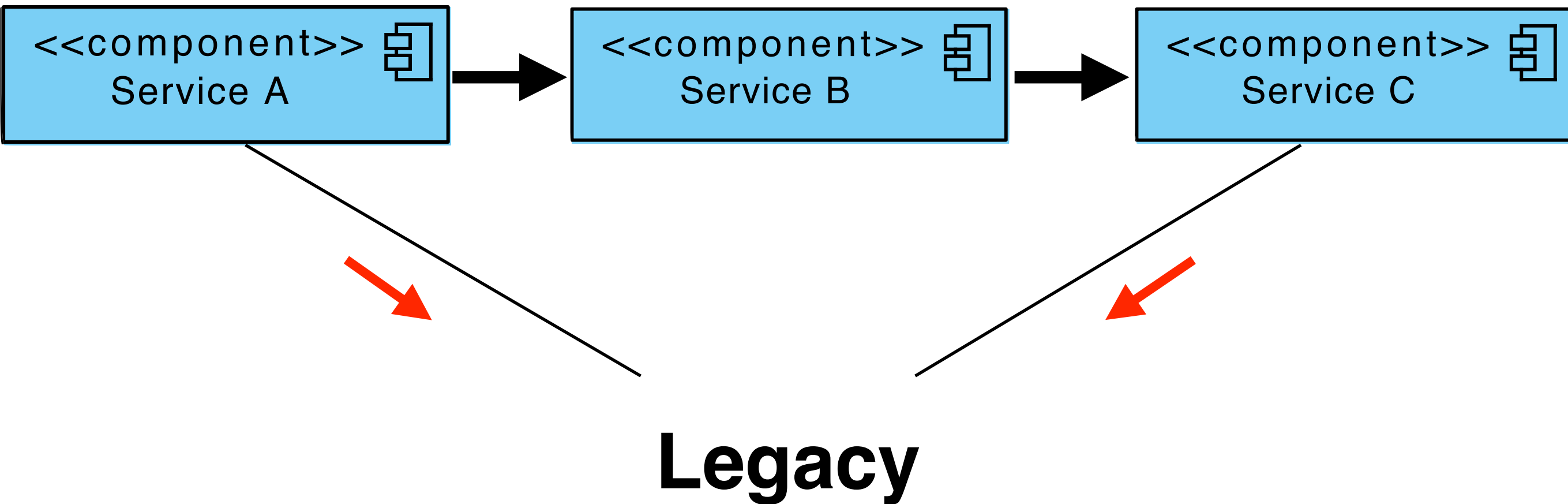


# Concept

## Process-oriented

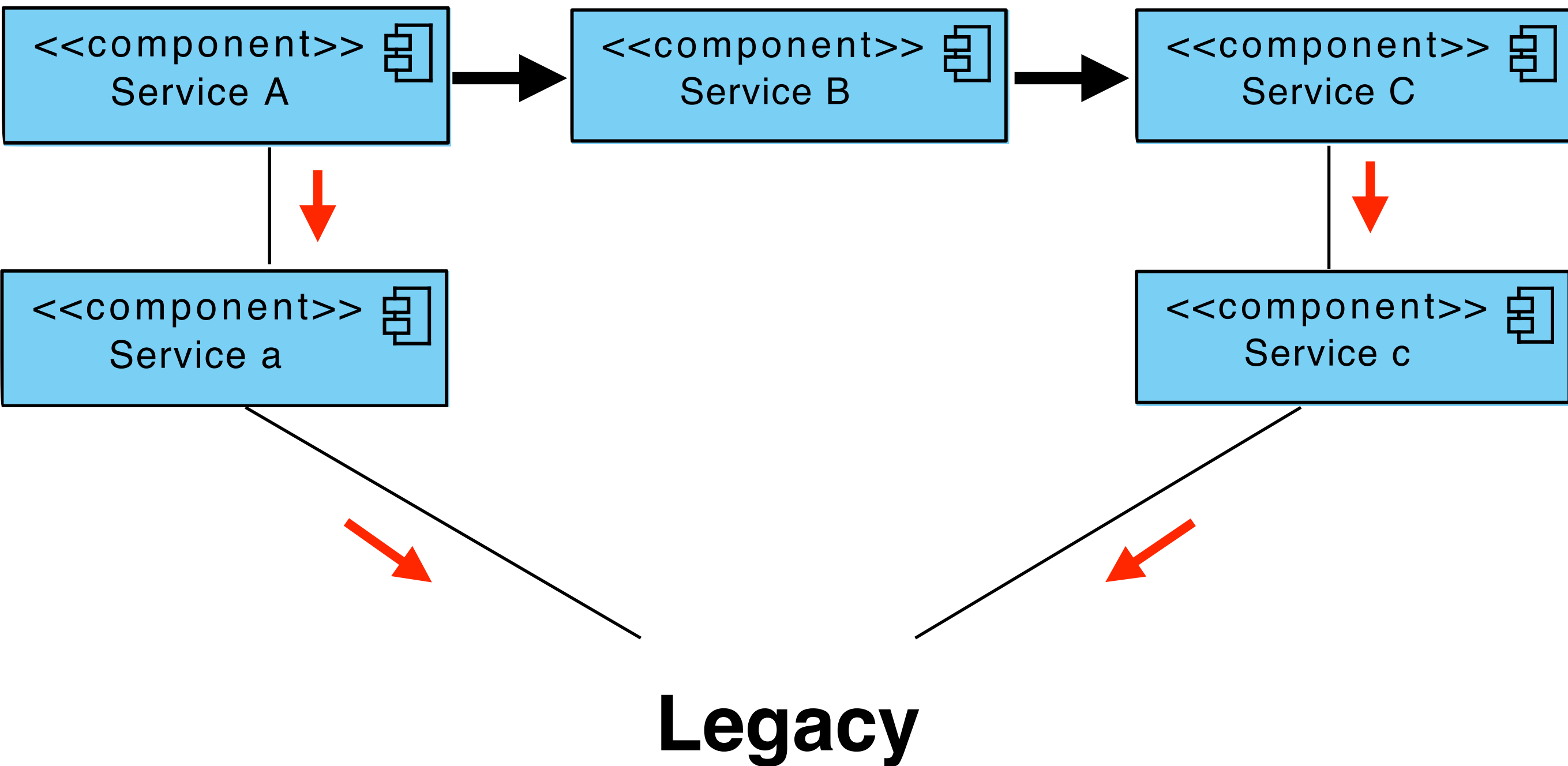


# Concept

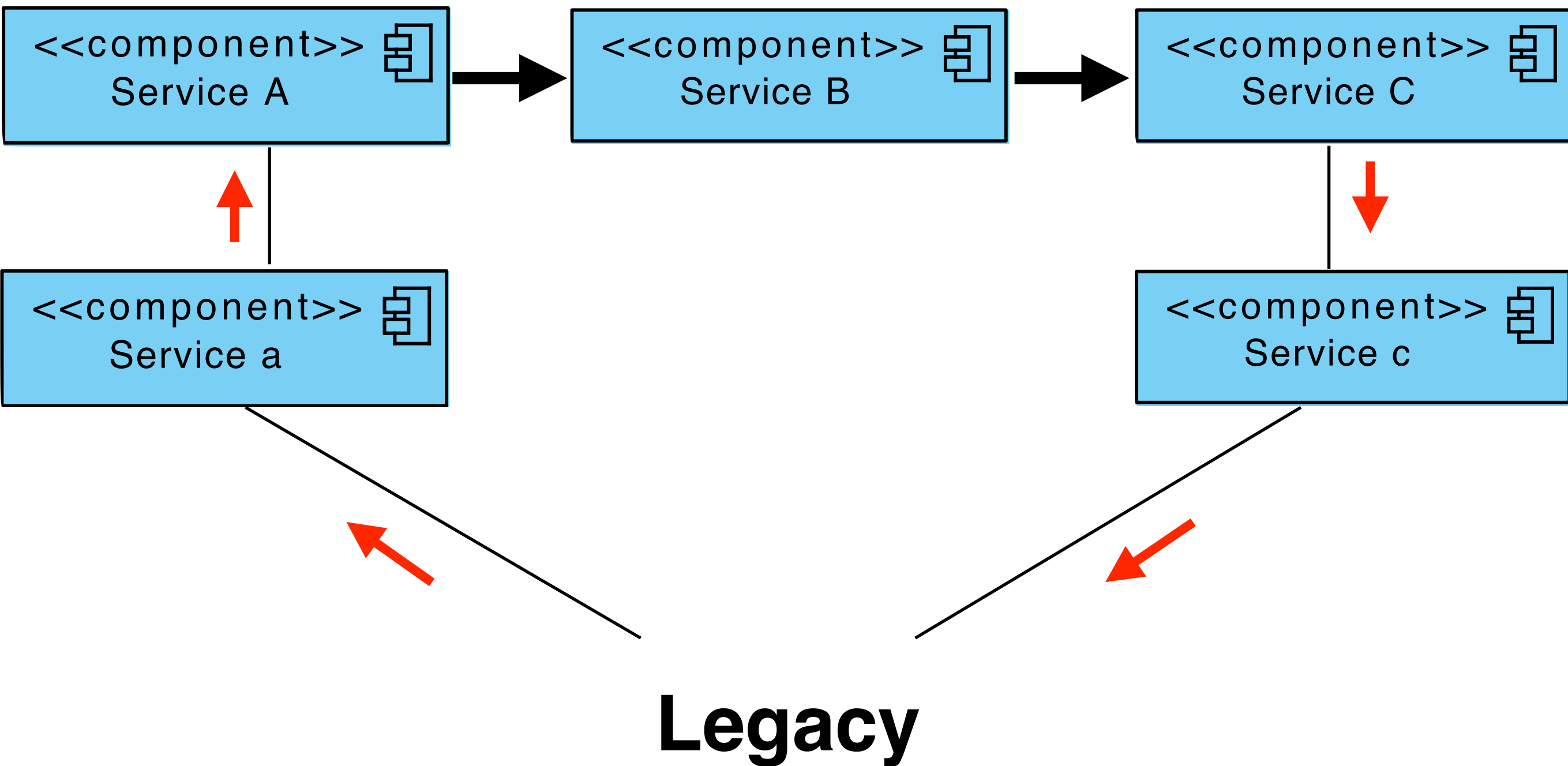




# Concept



# Concept



# Concept

## StranglerApplication



*Martin Fowler*

*29 June 2004*

When Cindy and I went to Australia, we spent some time in the rain forests on the Queensland coast. One of the natural wonders of this area are the huge strangler vines. They seed in the upper branches of a fig tree and gradually work their way down the tree until they root in the soil. Over many years they grow into fantastic and beautiful shapes, meanwhile strangling and killing the tree that was their host.

This metaphor struck me as a way of describing a way of doing a rewrite of an important system. Much of my career has involved rewrites of critical systems. You would think such a thing as easy - just make the new one do what the old one did. Yet they are always much more complex than they seem, and overflowing with risk. The big cut-over date looms, the pressure is on. While new features (there are always new

# Concept

project. They aren't yet at the point where the old application is strangled - but they've delivered valuable functionality to the business that gives the team the credibility to go further. And even if they stop now, they have a huge return on investment - which is more than many cut-over rewrites achieve.

The most important reason to consider a strangler application over a cut-over rewrite is reduced risk. A strangler can give value steadily and the frequent releases allow you to monitor its progress more carefully. Many people still don't consider a strangler since they think it will cost more - I'm not convinced about that. Since you can use shorter release cycles with a strangler you can avoid a lot of the unnecessary features that cut over rewrites often generate.

There's another important idea here - when designing a new application you should design it in such a way as to make it easier for it to be strangled in the future. Let's face it, all we are doing is writing tomorrow's legacy software today. By making it easy to be strangled in the future, you are enabling the graceful fading away of today's work.





# REFACTORING

IMPROVING THE DESIGN  
OF EXISTING CODE

**MARTIN FOWLER**

With Contributions by **Kent Beck, John Brant,  
William Opdyke, and Don Roberts**

Foreword by **Erich Gamma**  
Object Technology International Inc.



Robert C. Martin Series



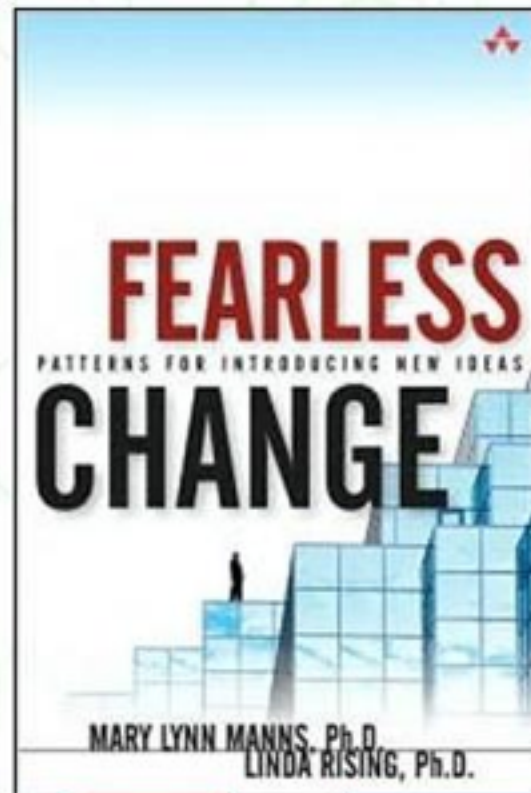
# WORKING EFFECTIVELY WITH LEGACY CODE

Michael C. Feathers



# FEARLESS CHANGE

patterns for introducing new ideas



Ask  
for Help

Time for  
Reflection

Plant  
the Seeds

Test the  
Waters

Bridge -  
Builder

**Have you ever tried to introduce a new idea into your organization?**

Even though change is difficult, leaders can't avoid it.

So wouldn't it be wonderful if people who have successfully introduced a new idea into an organization could share their stories with you?

Our books, *Fearless Change: Patterns for Introducing New Ideas* (2005) and *More Fearless Change: Strategies for Making Your Ideas Happen* (2015) are the next best thing.

**amazon**

[Order Fearless Change Book](#)  
[Order More Fearless Change Book](#)

**Thank you for your  
attention**