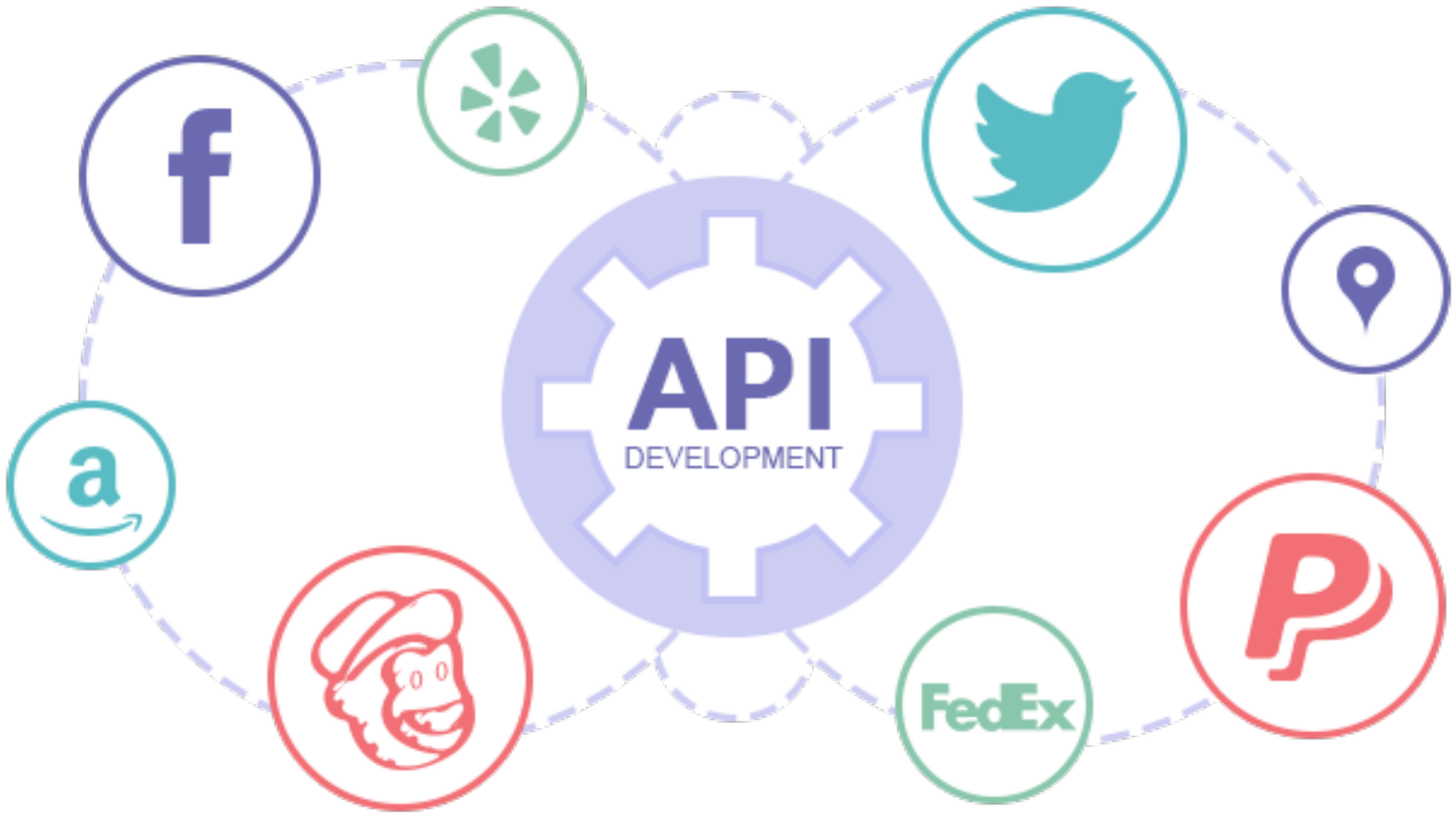
APIs (not only) in SAP environments

Martin Steinberg
@SbgMartin

Damir Majer
@majcon

API?



more and more companies expose their application‘s functionality to the outside world

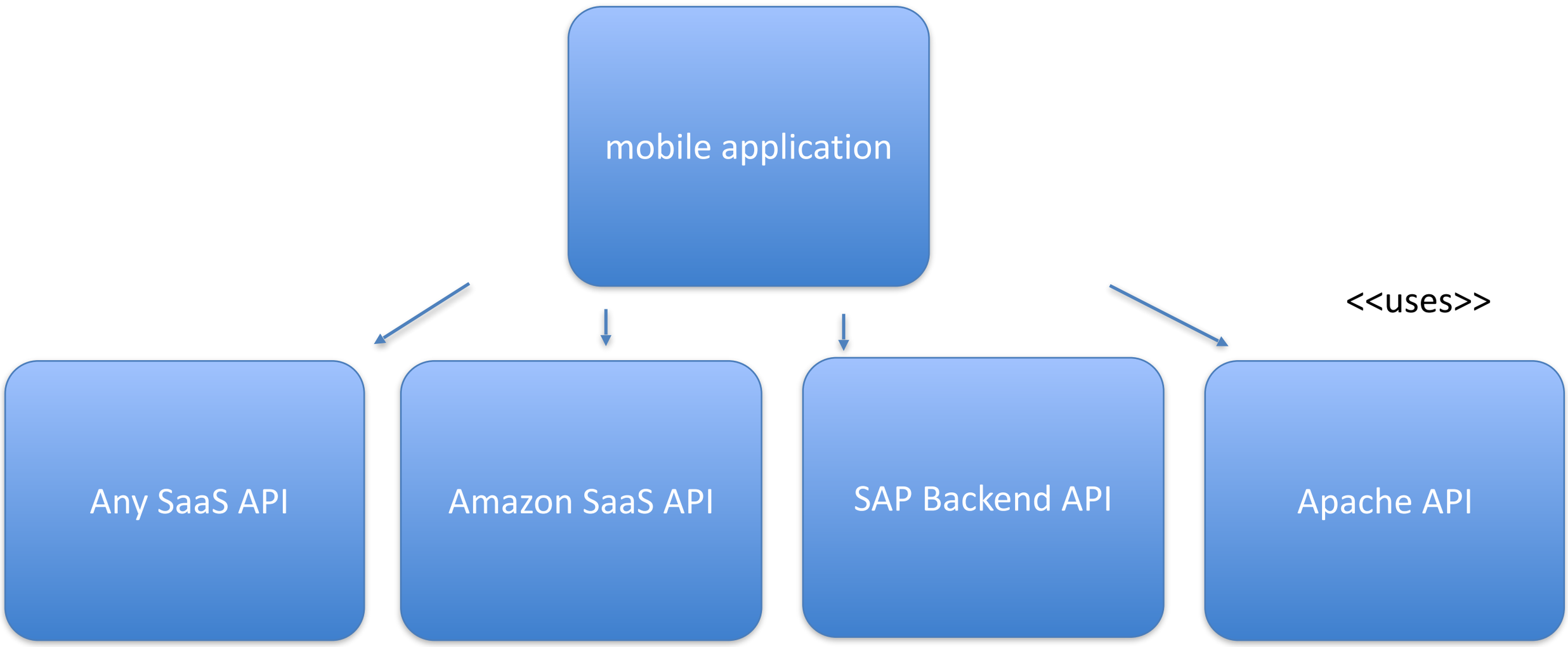
Copyright: http://www.webplanex.com/wp-content/uploads/2014/03/API-Development.png

Why is it important for me?

- API can be stated to be a foundation for building applications
- Innovative applications based on existing implementations/functionality
- Applications need functionality from different (cloud) services/systems
- API „customers“ are developers
- can be composed creatively

An example

- more and more applications use this exposed functionality



Why is it important for me?

- API‘s are usually available over internet (public)
- customers/users have to signup for usage („apikey“)
- „private API‘s“ for internal use (e.g. SAP backend system or SAP HANA system) within company‘s environment

7

Scalable

Simple

Security

Documentation

Versioning

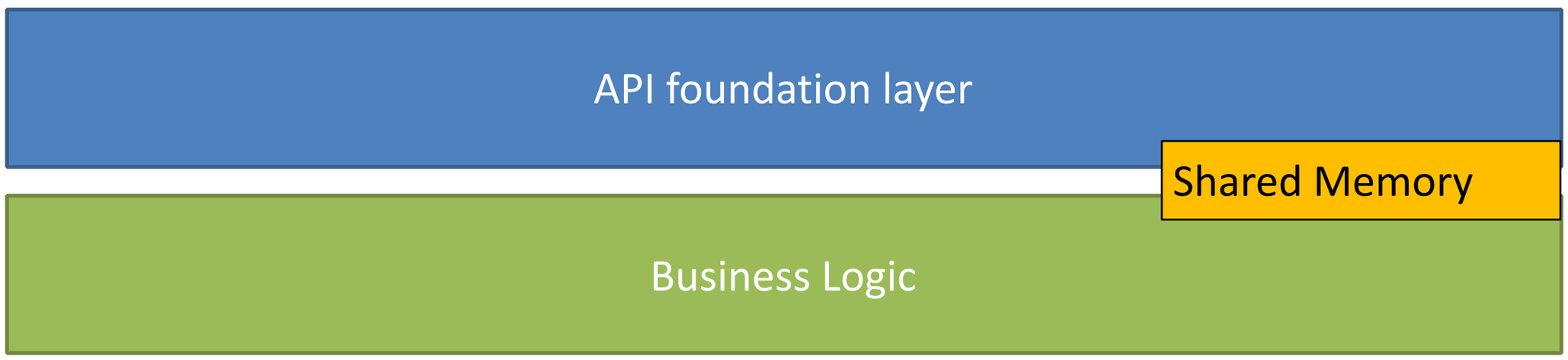
Analyzable

Monetizable

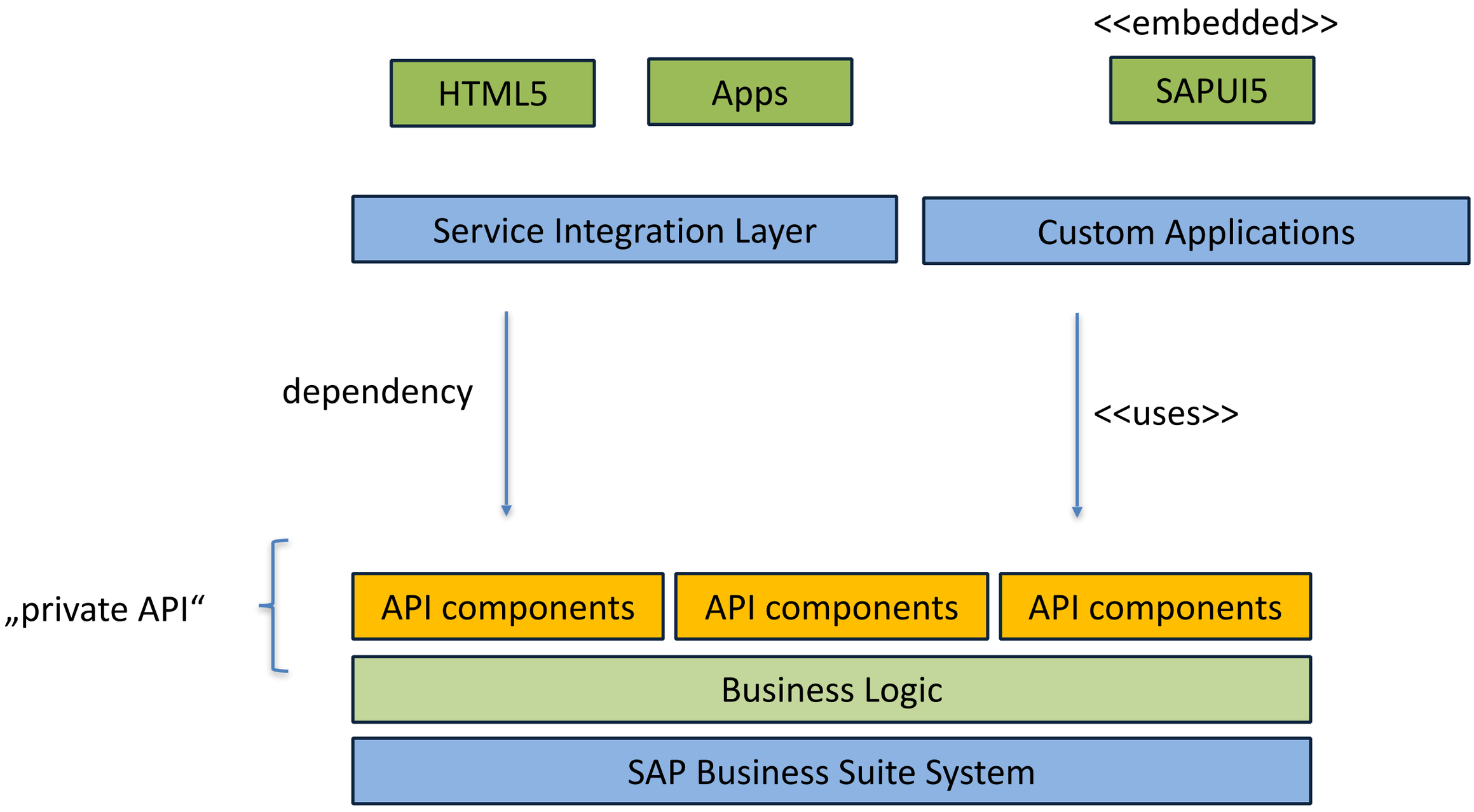
API foundation

Standardization of

- user Management
- authorization checks
- logging
- use of harmonized patterns
- discovery
- documentation



„Private API“


<<embedded>>
HTML5
Apps
SAPUI5
Service Integration Layer
Custom Applications
dependency
<<uses>>
„private API“
API components
API components
API components
Business Logic
SAP Business Suite System


THINKER
ThinKing Cap
IDEAOMETER



- Systems should expose behavior with interfaces
- Interfaces are meant and not Classes…
- Interfaces only show the Specification and allow clear isolation from the implementation part

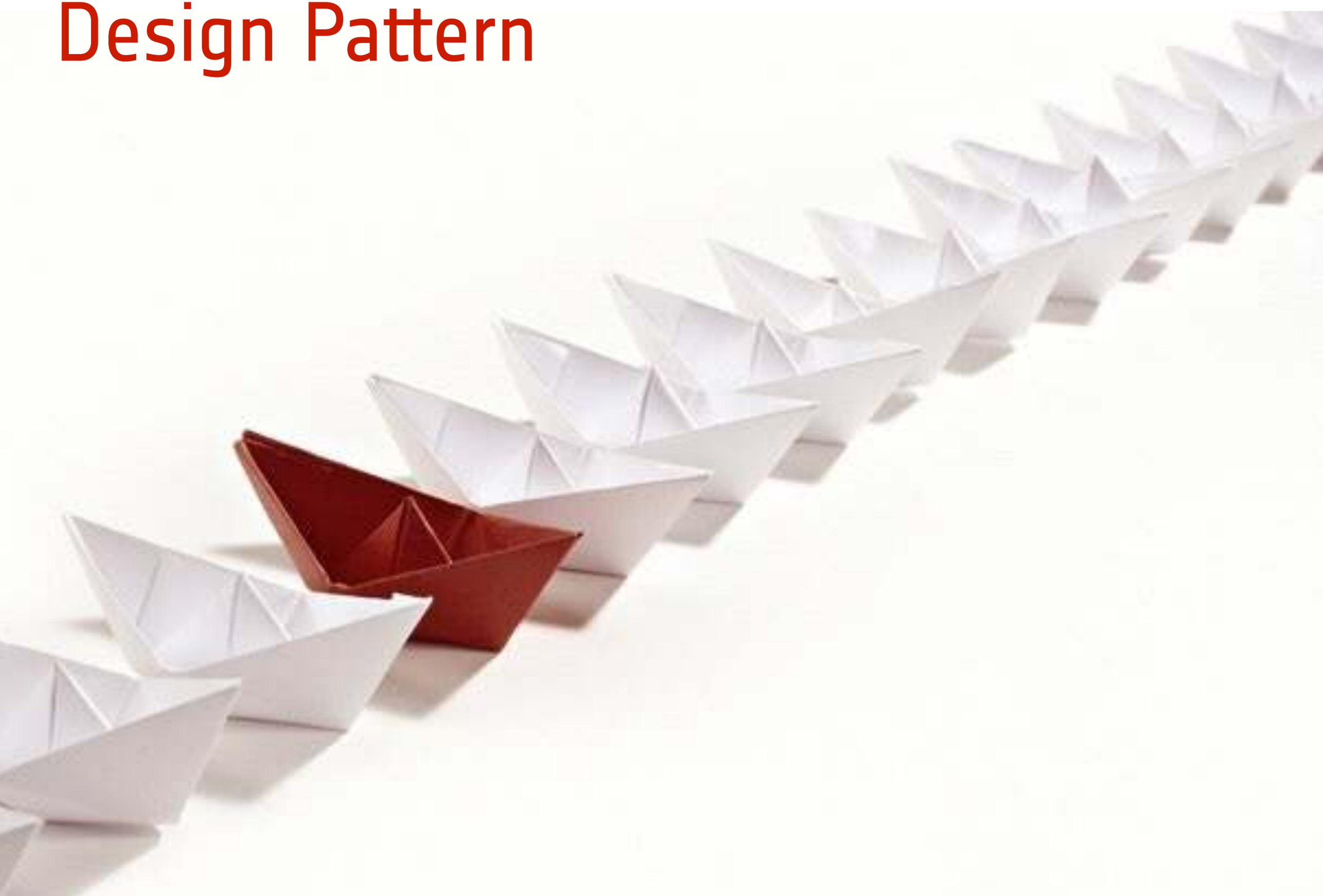



```
class lcl_very_important definition.
public section.
    interfaces: lif_smart_interface.
    "…
endclass.

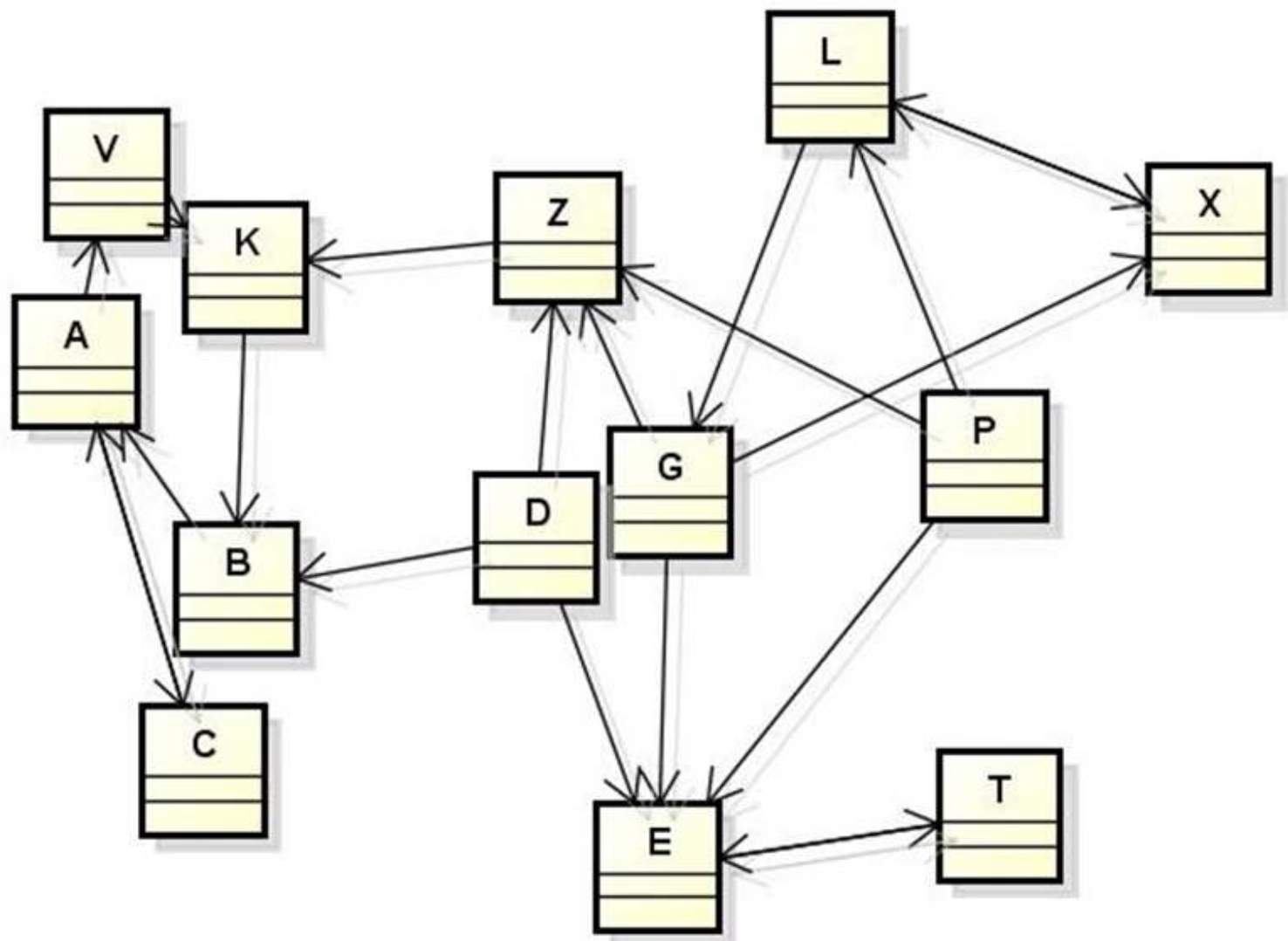
class lcl_very_important implementation.
    "…
    method lif_smart_interface~applause.
    " really cool and fancy implementations
    " it´s your action ;-)
    endmethod.
endclass.
```

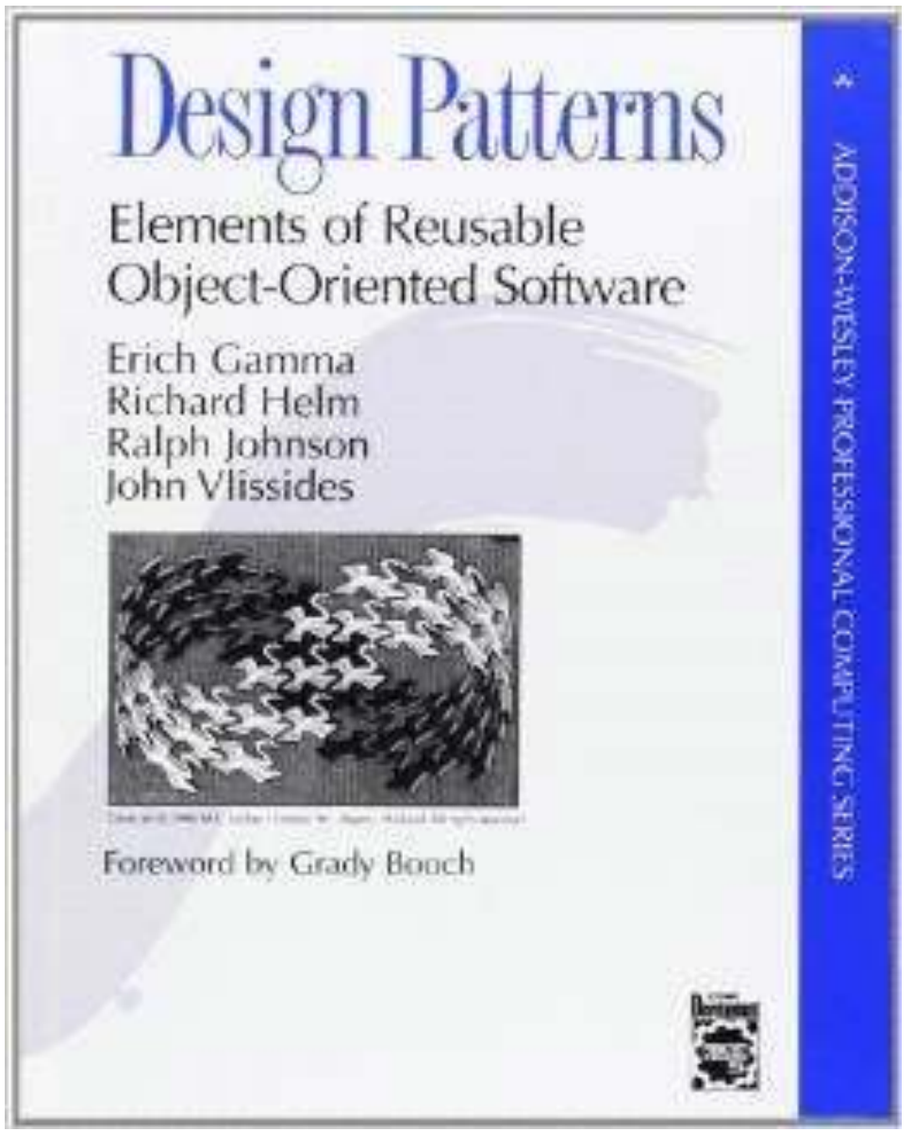
```
interface lif_smart_interface.
    public section.
        methods: applause
            importing iv_loudness type i.
endinterface.
```

Design Pattern






L
V
K
Z
X
A
P
G
D
B
C
E
T



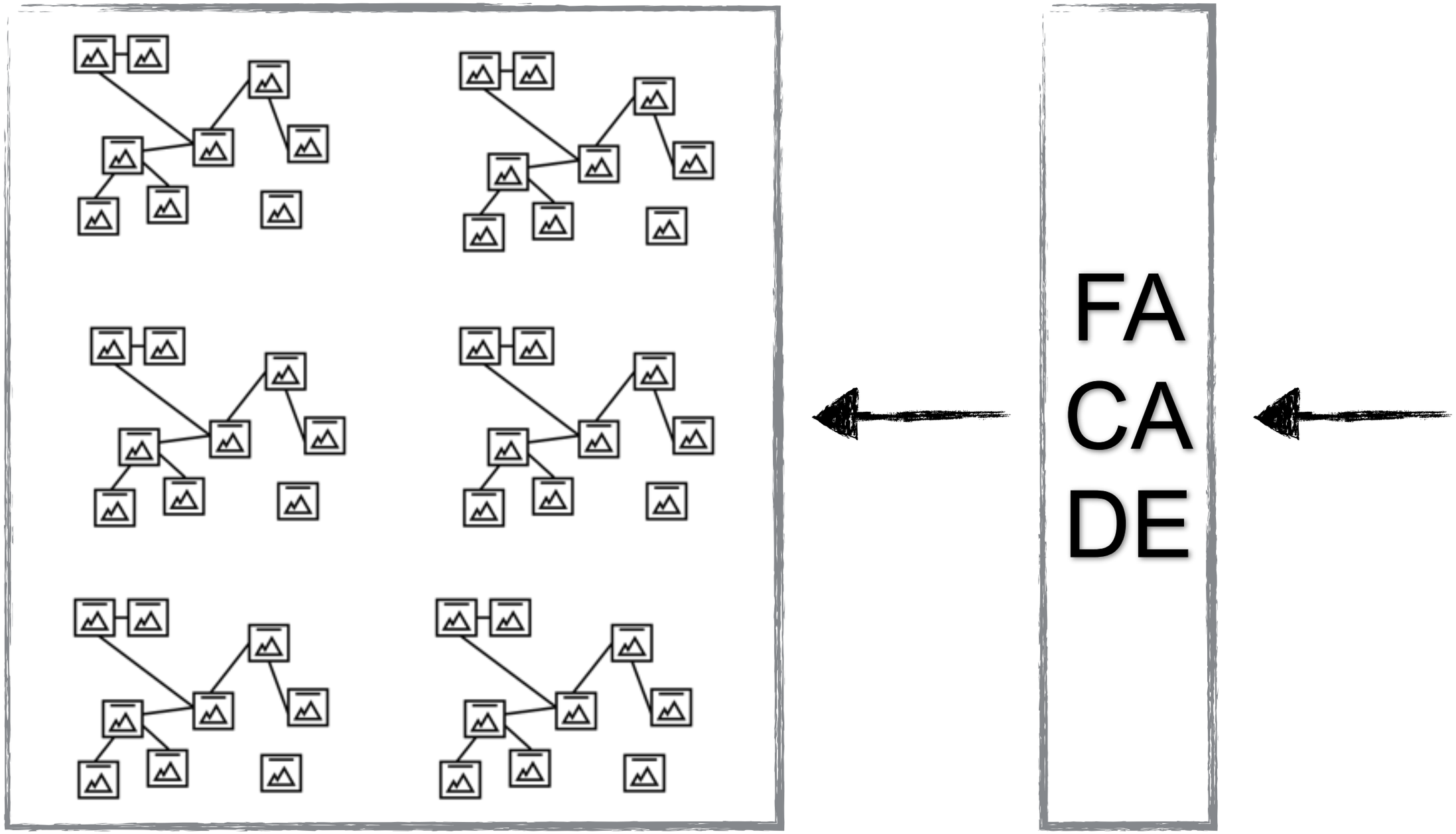
Pattern categories



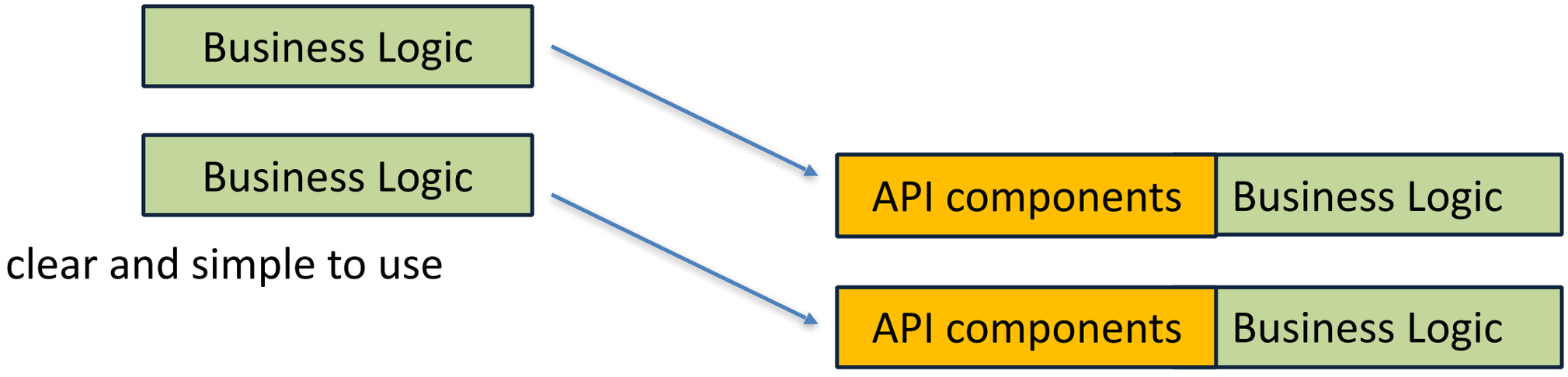
creational structural behavioral

- complex Solutions are defined by a Name
- Developer has an unique vocabulary

Facade Pattern



API as a facade



clear and simple to use



delivered or custom

Idealistic approach:
harmonized object model
homogenous interface design

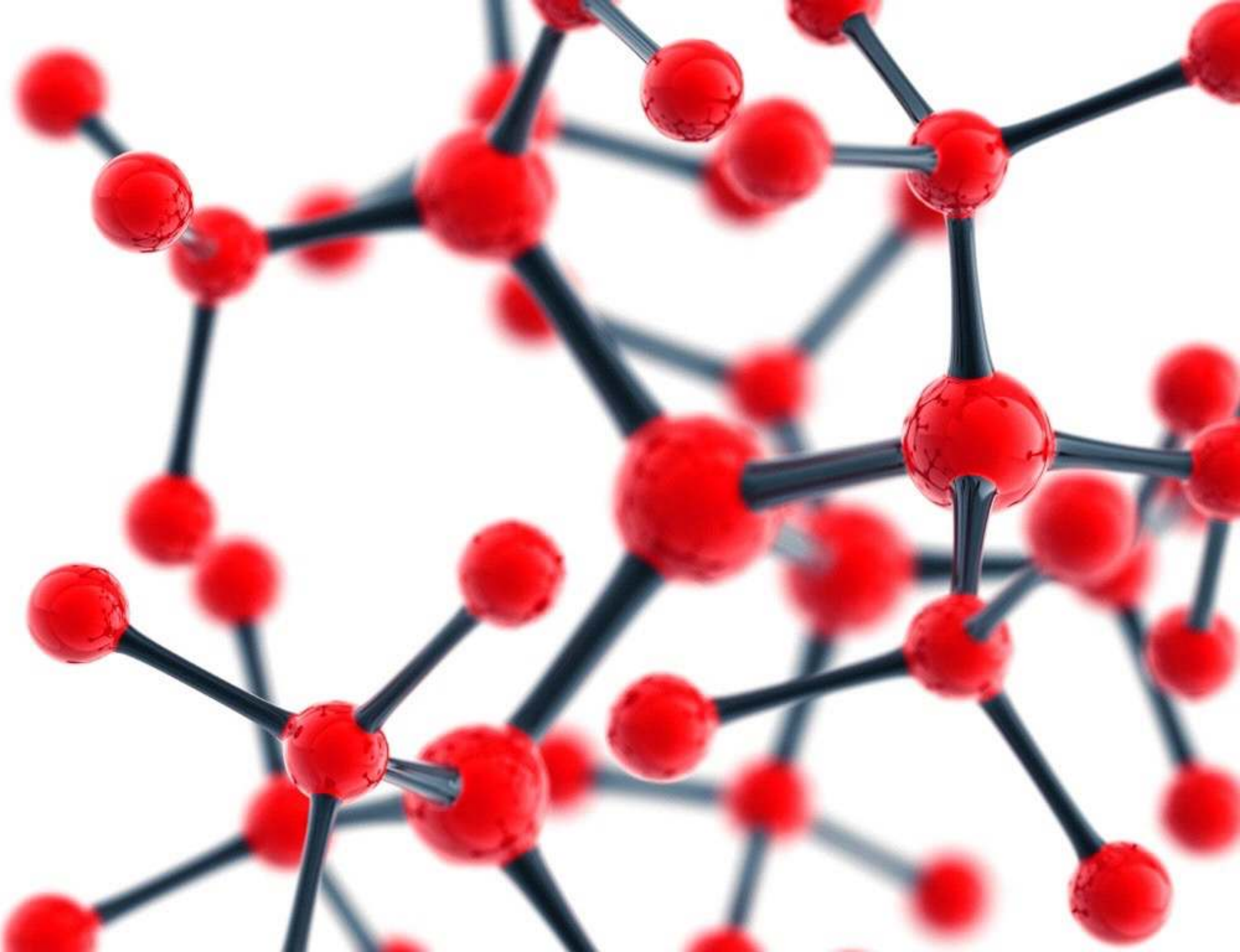
API components – from a quality perspective

Let‘s differentiate between 3 different quality levels:

- Tools
  - for time-limited usage, like data migration report
- Potential library components
  - any implementation which is not library, yet
- Library components
  - automated testing necessary from the beginning
  - stable interface
  - Versioning
  - Documentation

- API components are „library components“

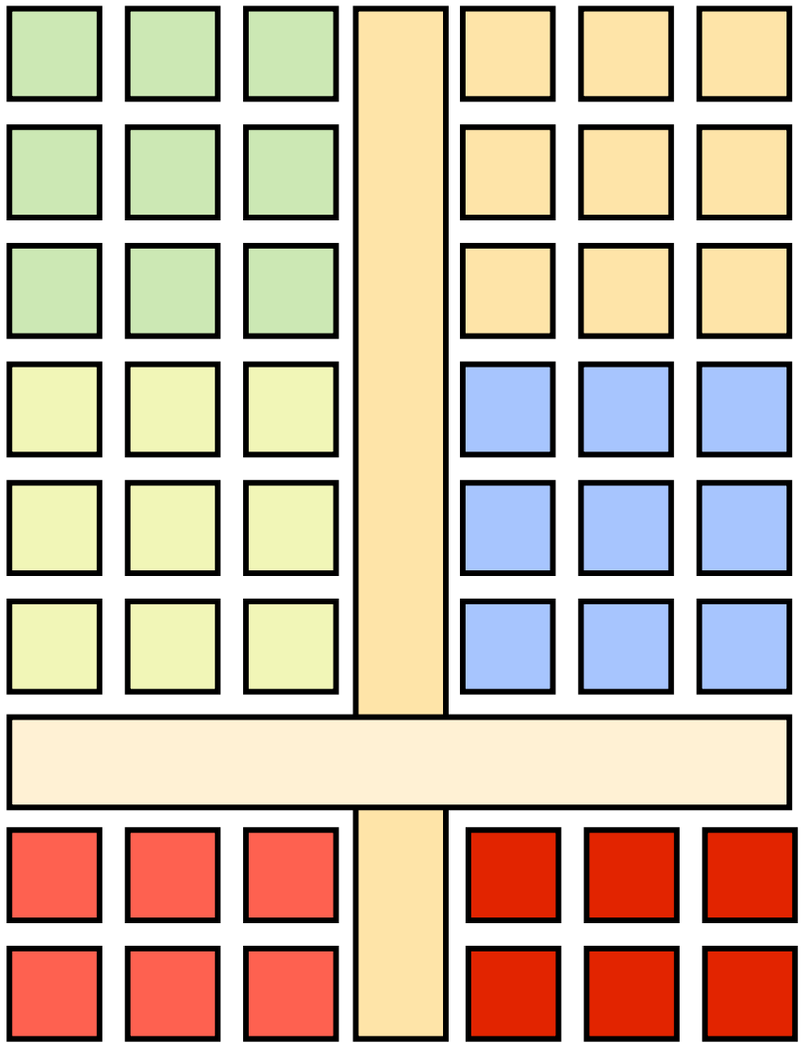
- ABAP Unit
- Documentation (ABAP Doc + DMS)

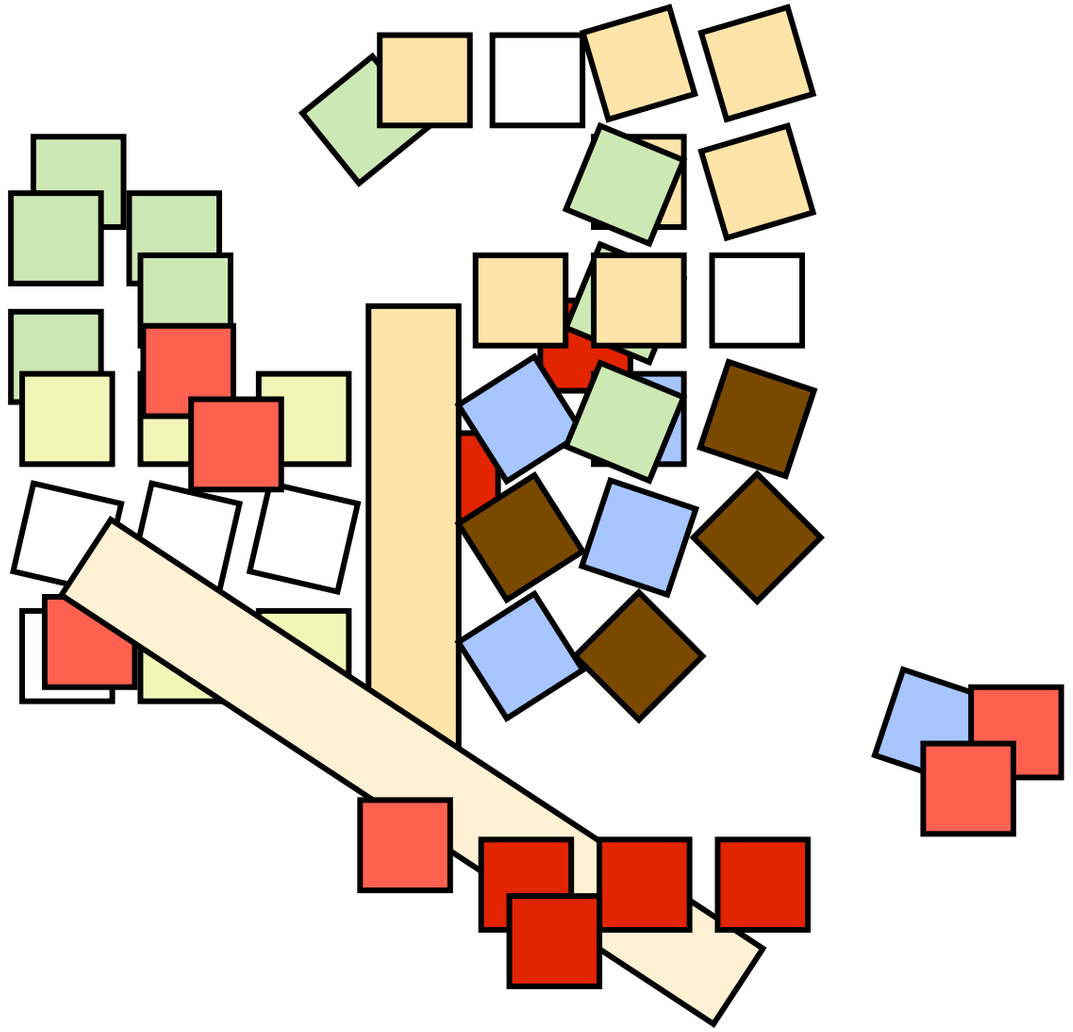




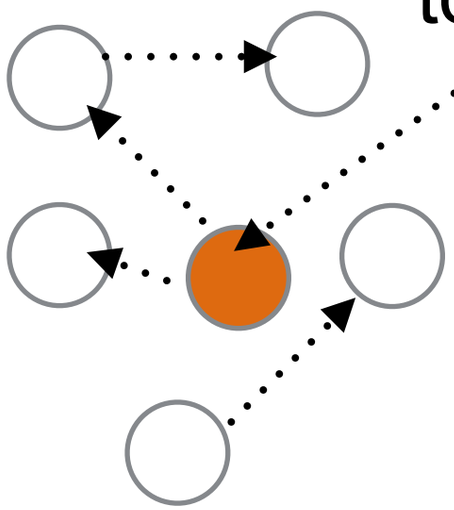
25% - 50% of Development costs are used for Testing

Software should be simple tested

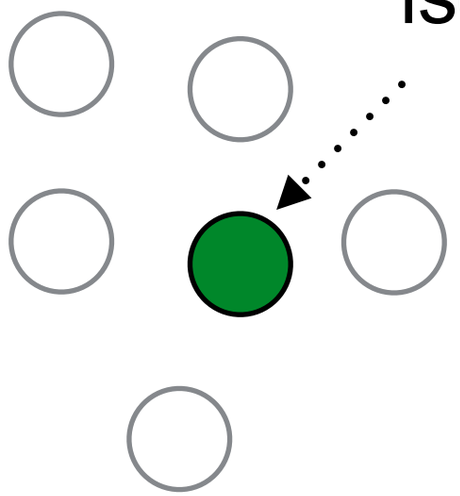


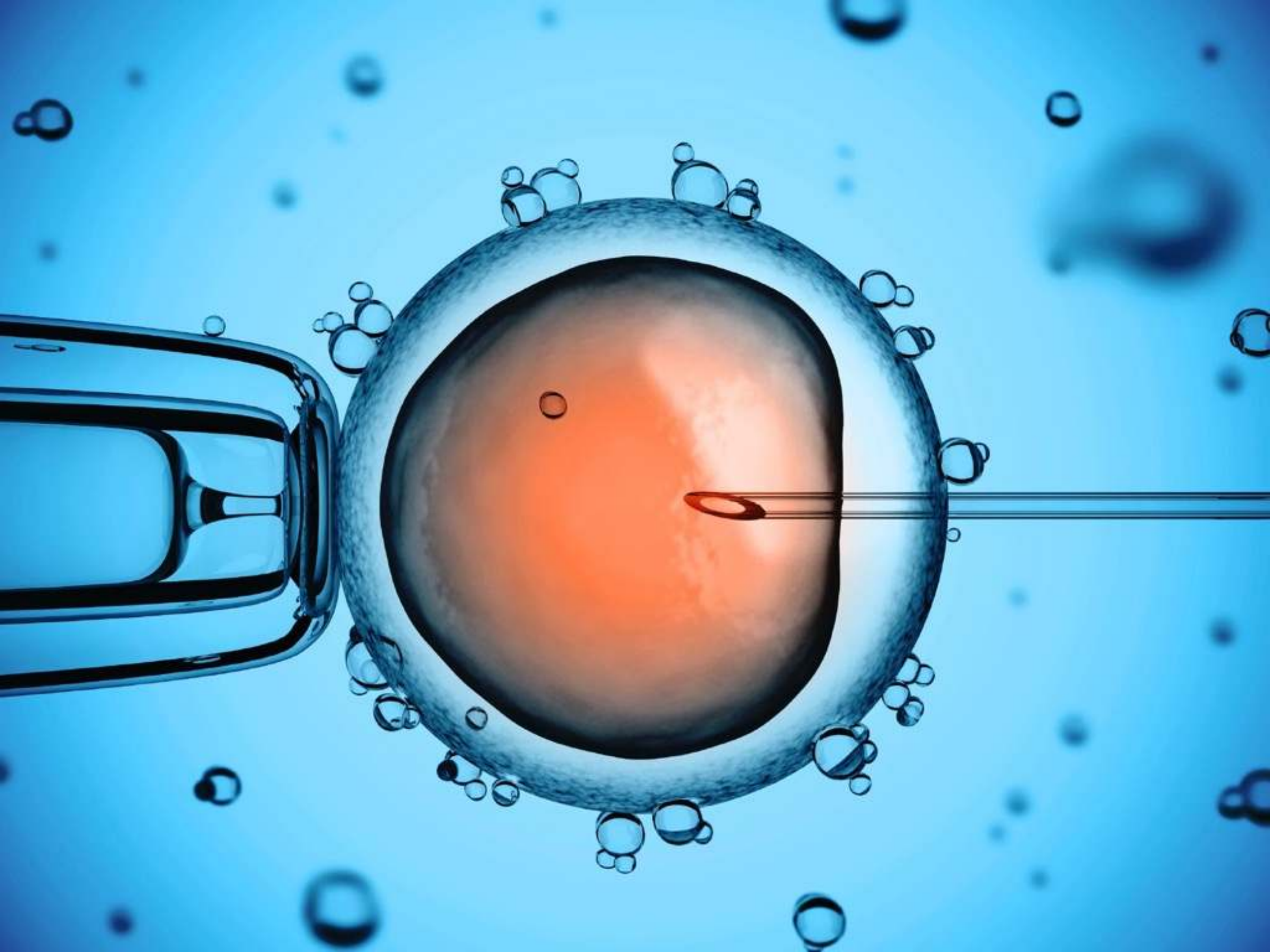


too much dependencies



isolated tested





Dependency Injection

- concept idea by Martin Fowler
- based on Single Responsibility Principle (SRP)

- a class uses available functionality of another class only by implementing against the interface
- allows different implementation approaches

http://en.wikipedia.org/wiki/Dependency_injection
http://en.wikipedia.org/wiki/Dependency_inversion_principle
http://en.wikipedia.org/wiki/SOLID_%28object-oriented_design%29

Dependency Injection

```
"! interface representing a subsystem
INTERFACE lif_google_address_api_proxy.
  METHODS:
    get_geocoordinates RETURNING VALUE(rs_geo_info) TYPE ty_sitfra_geo_info.
ENDINTERFACE.

"! Our API implementation class
CLASS lcl_customer_api_impl_1 DEFINITION.
  PUBLIC SECTION.
    METHODS constructor IMPORTING ir_geocode_impl TYPE REF TO lif_google_address_api_proxy OPTIONAL.
    METHODS get_customer_address RETURNING VALUE(rs_address) TYPE adrc.

  PROTECTED SECTION.
  PRIVATE SECTION.
    DATA:
      "! member reference for Google geocode API
      mr_google_geo_api TYPE REF TO lif_google_address_api_proxy.
ENDCLASS.

CLASS lcl_customer_api_impl_1 IMPLEMENTATION.

  METHOD constructor.
  METHOD get_customer_address.

    DATA(lt_geo) =  mr_google_geo_api->get_geocoordinates( ).
  ENDMETHOD.

ENDCLASS.
```

Isolation by mocking

- Mocking is a principle coming from test driven development
- use dependency injection for mocking purposes
- use it for contract-first approach

```
CLASS lcl_google_mocker DEFINITION.

  PUBLIC SECTION.
    INTERFACES lif_google_address_api_proxy.
  PROTECTED SECTION.
  PRIVATE SECTION.

ENDCLASS.

CLASS lcl_google_mocker IMPLEMENTATION.

  METHOD lif_google_address_api_proxy~get_geocoordinates.
    "return static value -> predictable result of this method
    rs_geo_info-h_value = |5544718.652|.
    rs_geo_info-r_value = |3468709.067|.
  ENDMETHOD.

ENDCLASS.

CLASS ltcl_customer_api_impl_1 DEFINITION FINAL FOR TESTING
  DURATION SHORT
  RISK LEVEL HARMLESS.

  PRIVATE SECTION.
    DATA:
        mr_customer_api_1 TYPE REF TO lcl_customer_api_impl_1.
    METHODS:
      setup,
      get_customer_address_01 FOR TESTING RAISING cx_static_check.
ENDCLASS.


CLASS ltcl_customer_api_impl_1 IMPLEMENTATION.

  METHOD get_customer_address_01.

    DATA(lt_geo) = mr_customer_api_1->get_customer_address( ).

  ENDMETHOD.

  METHOD setup.

    mr_customer_api_1 = NEW #( NEW lcl_google_mocker( ) ).

  ENDMETHOD.
ENDCLASS.
```

Public API

HTML5 | Apps | Customers | …

„public API“

API components | API components | API components

API Mgmt.

e.g. AS ABAP
frontend server(s)
EAI

Service Integration Layer



„private API“

API components | API components | API components

Business Logic

SAP Business Suite System

Public API provider

API components

Service Integration Layer

API Mgmt.

Google, …

apigee

API components

Service Integration Layer

API Mgmt.

Hybris

SAP HANA Cloud Platform

Extension Apps

apigee

API components

Gateway as a Service

API Mgmt.

API components

Service Integration Layer

API Mgmt.

API components

Service Integration Layer

API Mgmt.

Did you recognize the yaas.io announcement?

SUCCESS



is to be flexible



Martin Steinberg

about.me/martin.steinberg
twitter: @SbgMartin



Damir Majer

www.majcon.de
twitter: @majcon

Thank you for your attention!

