Software Architecture Conference

Architecture.Next: Invalidating Old Axioms

Mark Richards

Independent Consultant

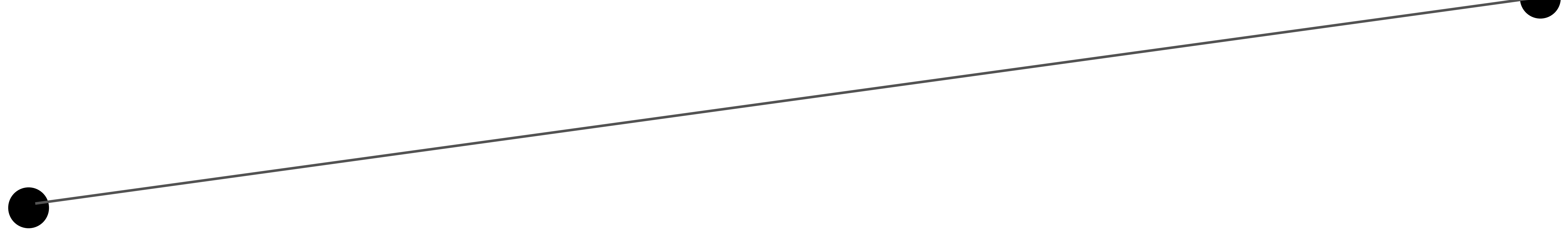
Hands-on Software Architect

Published Author|Conference Speaker

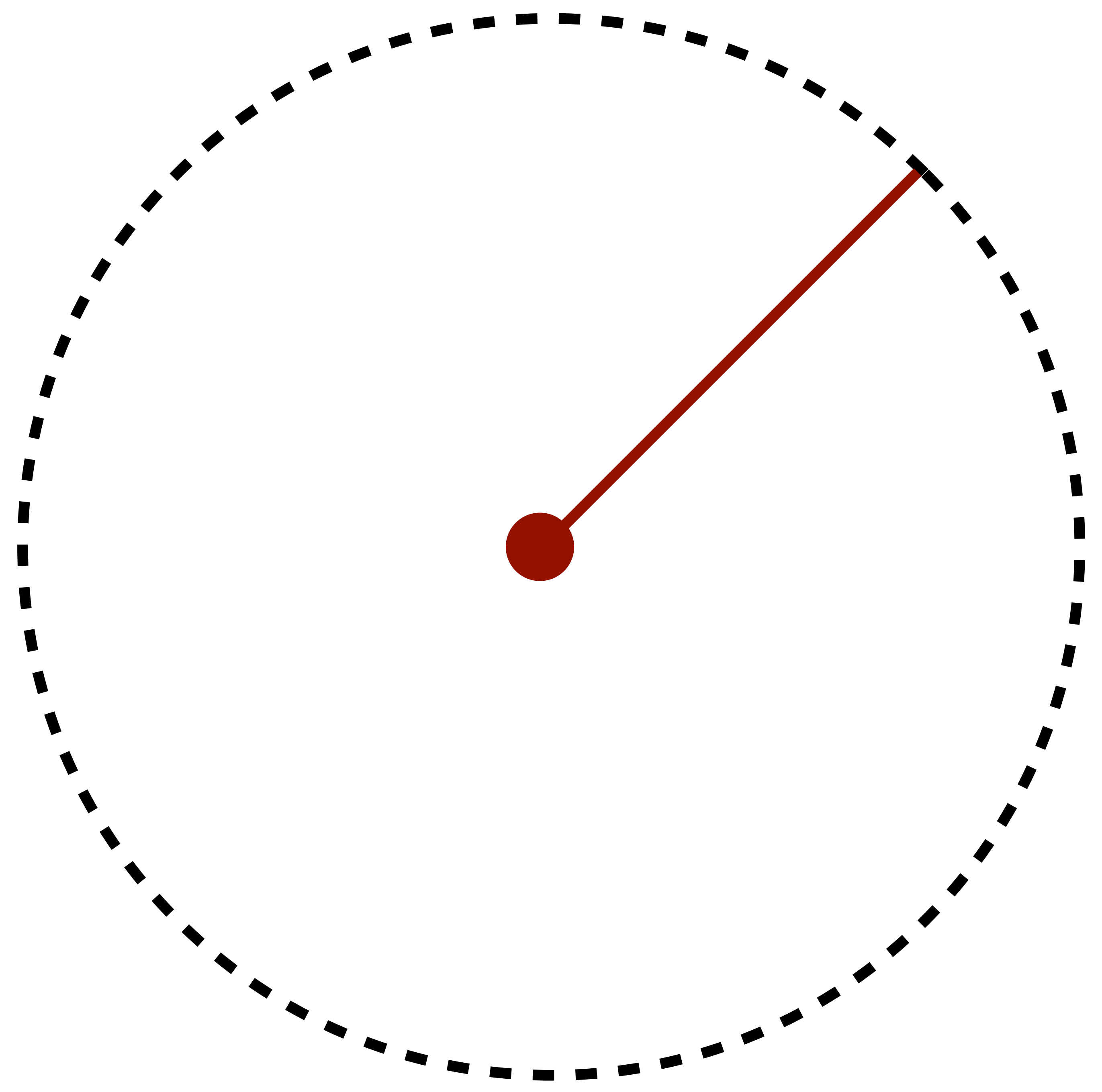
O'REILLY®

ax·i·om

a statement or proposition which is regarded as being established, accepted, or self-evidently true it is possible to draw a straight line from any point to any other point.



it is possible to describe a circle with any center and any radius

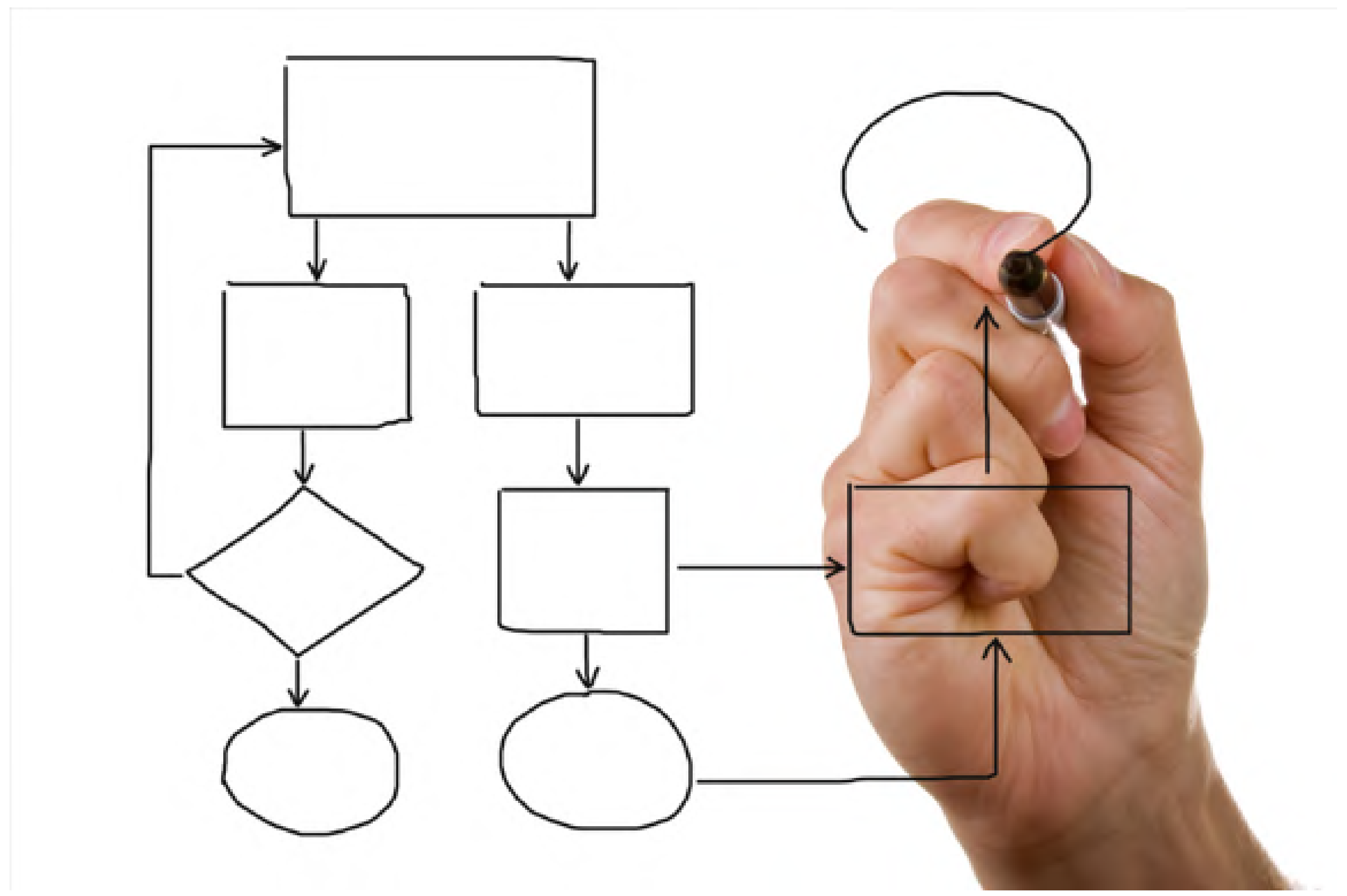


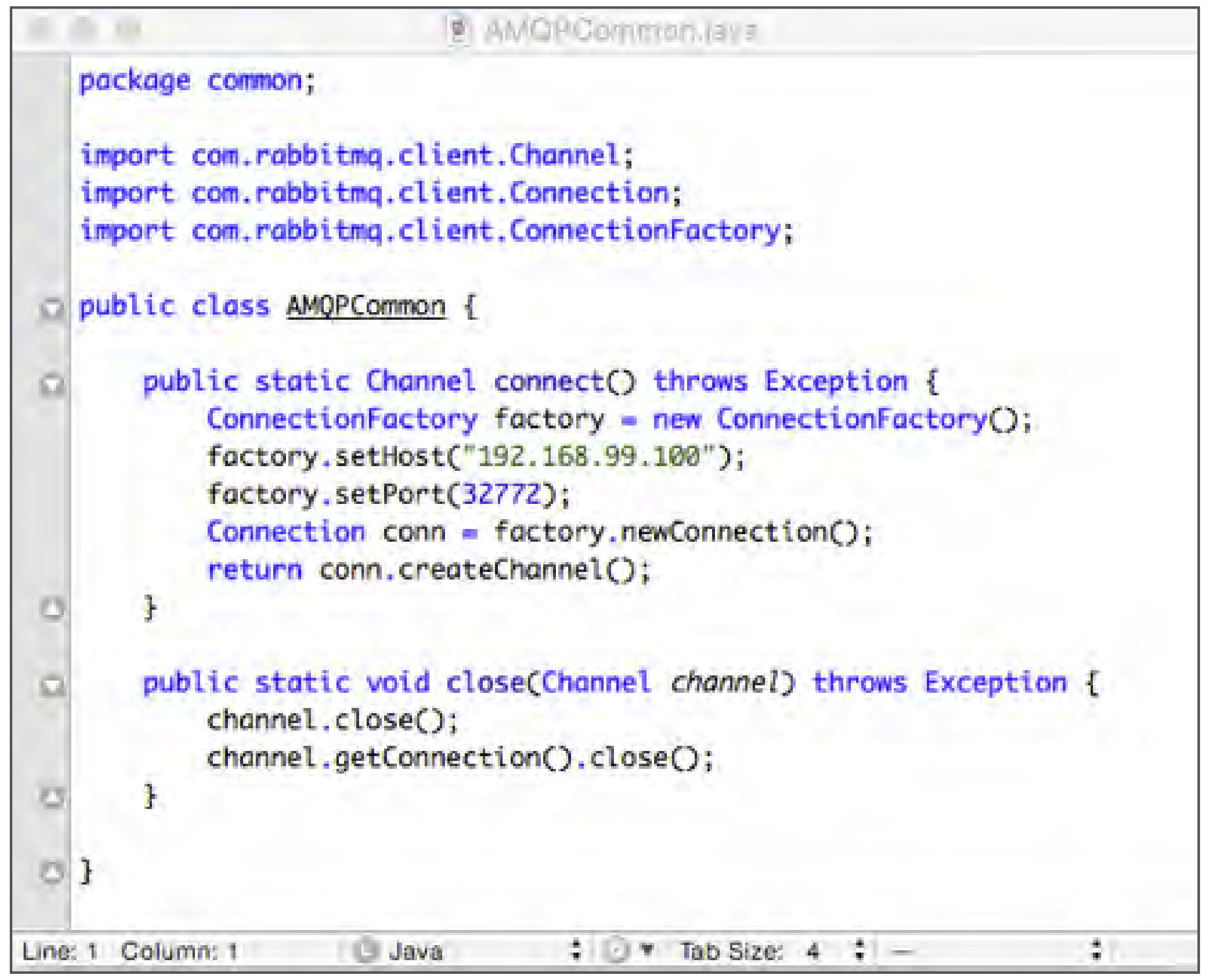
(axiom of extensionality)

given any set A and any set B, if for every set X, X is a member of A if and only if X is a member of B, then A is equal to B.

$$\forall A\,\forall B\,(\forall X\,(X \in A \iff X \in B) \implies A = B)$$

software architecture is a separate activity from software development

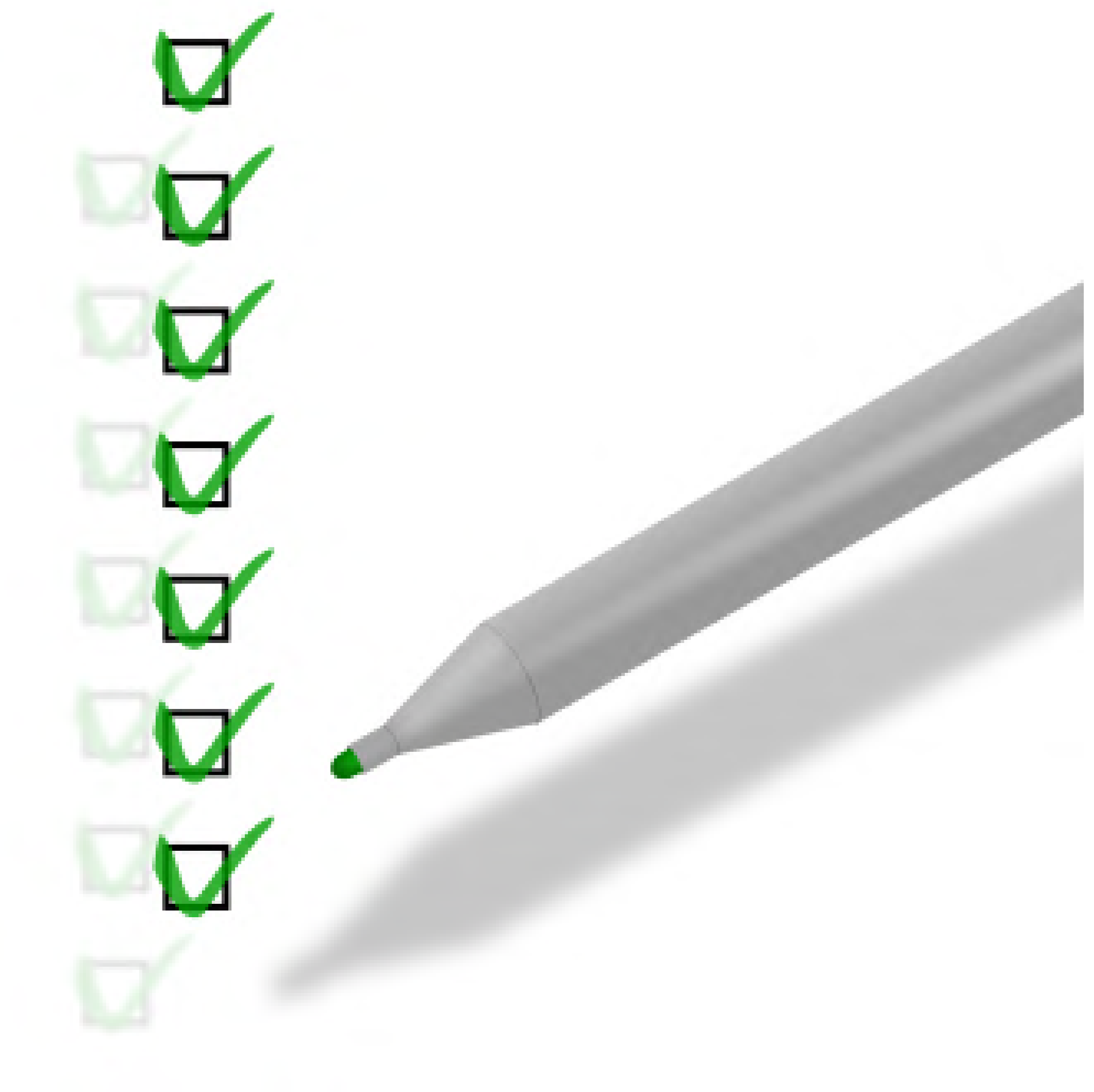




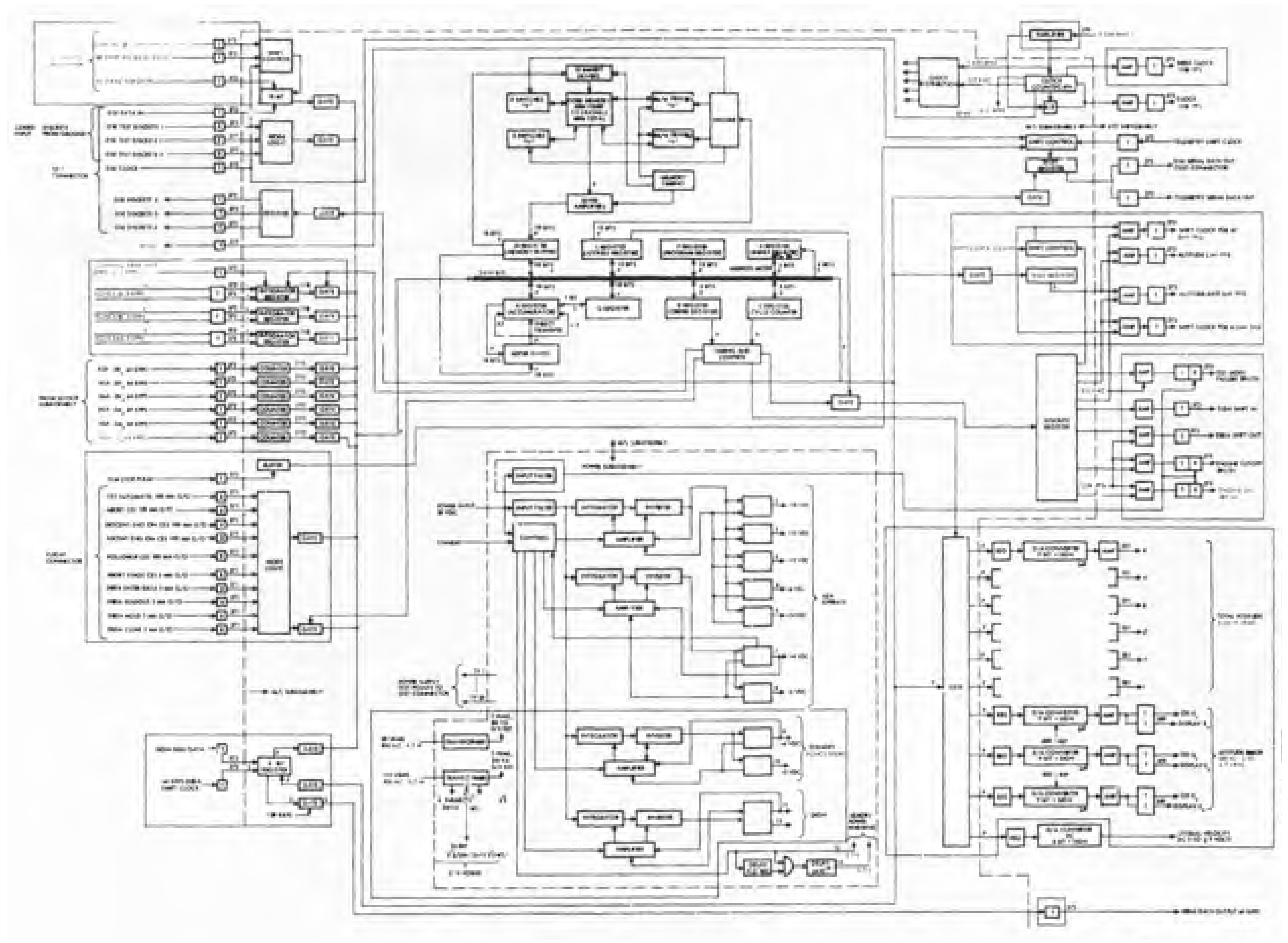
```
package common;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class AMQPCommon {

    public static Channel connect() throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("192.168.99.100");
        factory.setPort(32772);
        Connection conn = factory.newConnection();
        return conn.createChannel();
    }

    public static void close(Channel channel) throws Exception {
        channel.close();
        channel.getConnection().close();
    }

}
```

Line: 1 Column: 1 Java Tab Size: 4

software architects should adopt and follow best practices in software architecture

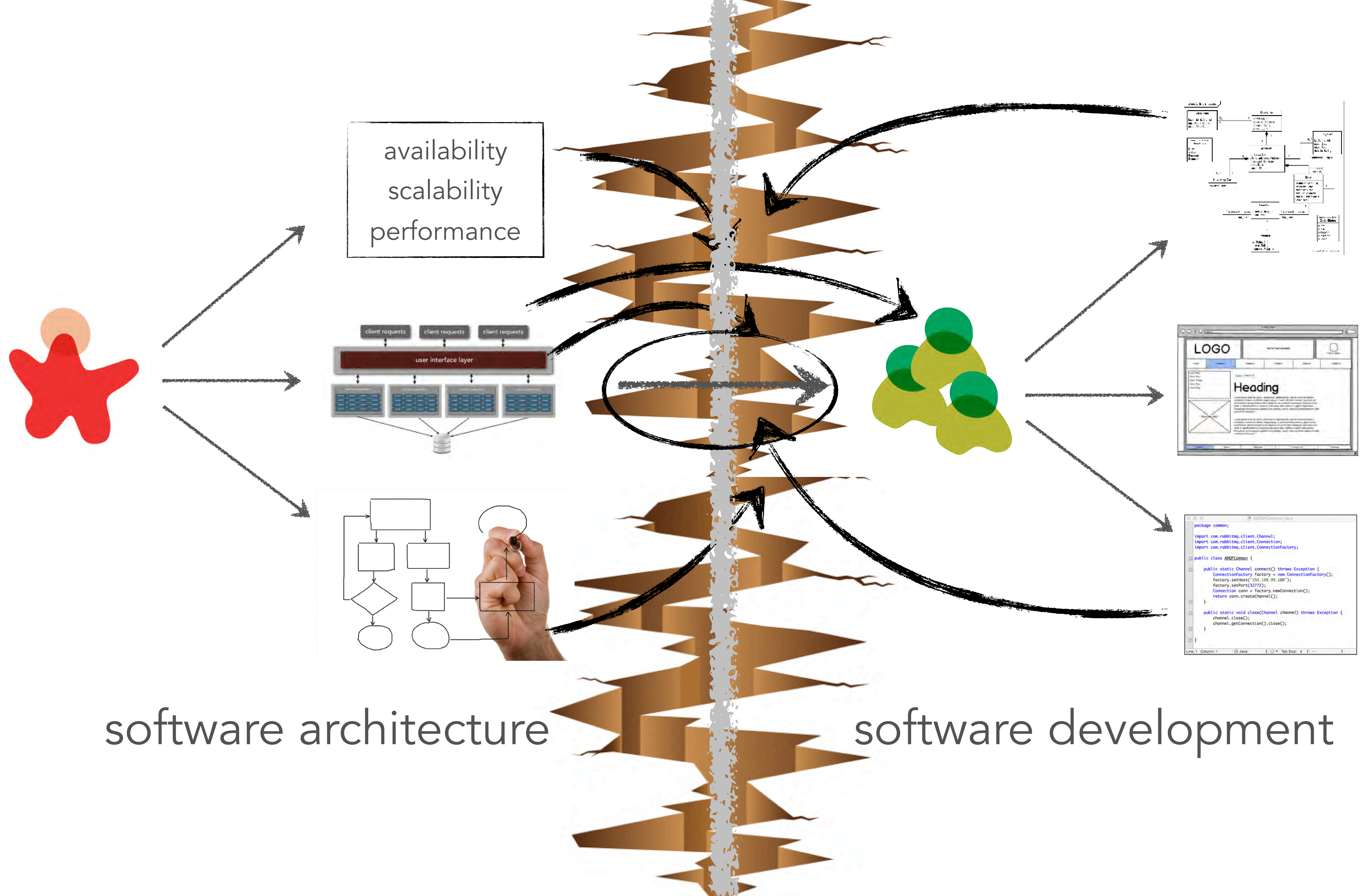


software architecture is the stuff that's hard to change later



software architecture is a separate activity from software development

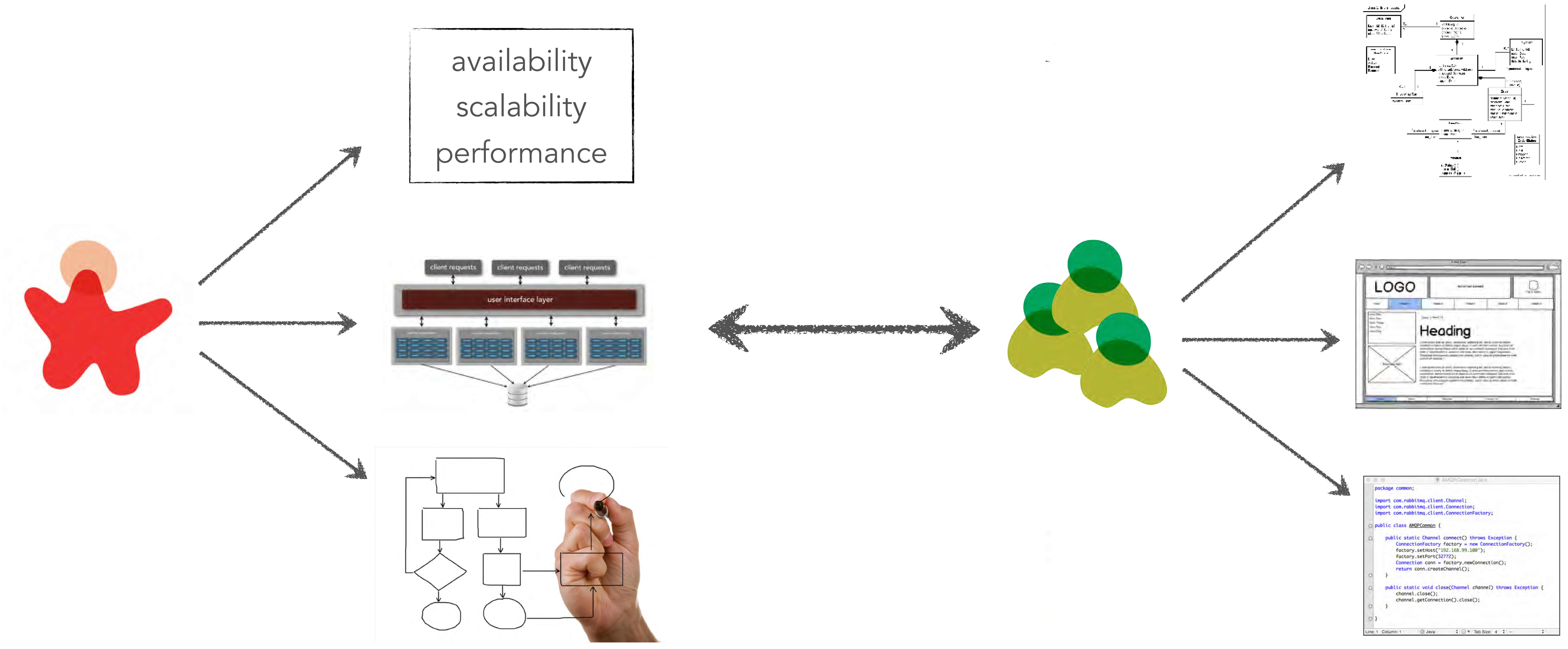
REJECTED



```java
package common;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class AMQPCommon {

    public static Channel connect() throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("192.168.99.100");
        factory.setPort(32772);
        Connection conn = factory.newConnection();
        return conn.createChannel();
    }

    public static void close(Channel channel) throws Exception {
        channel.close();
        channel.getConnection().close();
    }

}
```

software architecture

software development

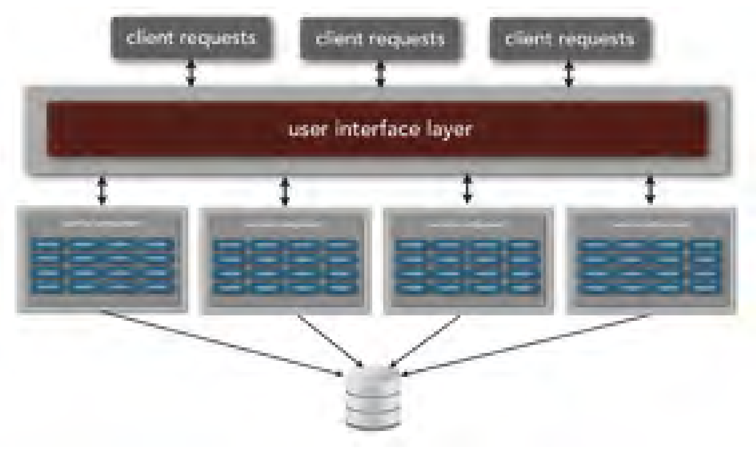


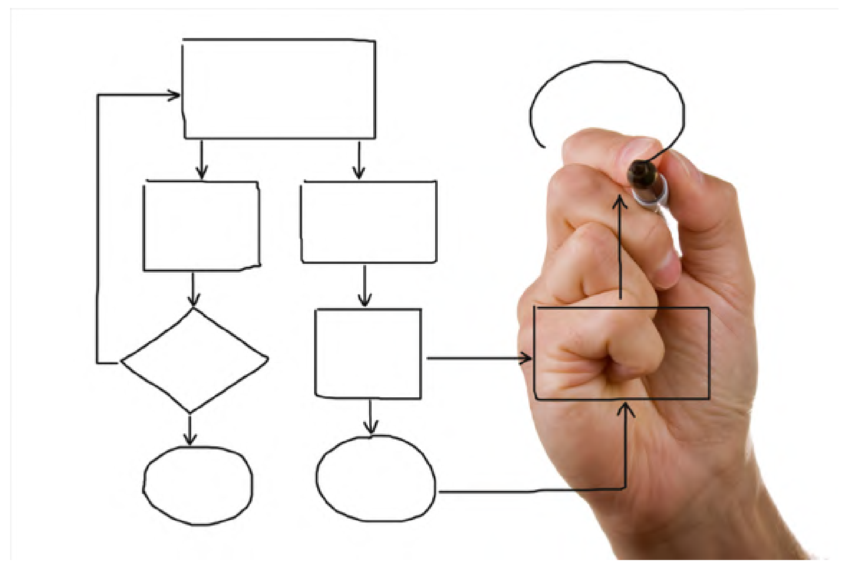
```
package common;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class AMQPCommon {

    public static Channel connect() throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("192.168.99.100");
        factory.setPort(32772);
        Connection conn = factory.newConnection();
        return conn.createChannel();
    }

    public static void close(Channel channel) throws Exception {
        channel.close();
        channel.getConnection().close();
    }

}
```

software architecture

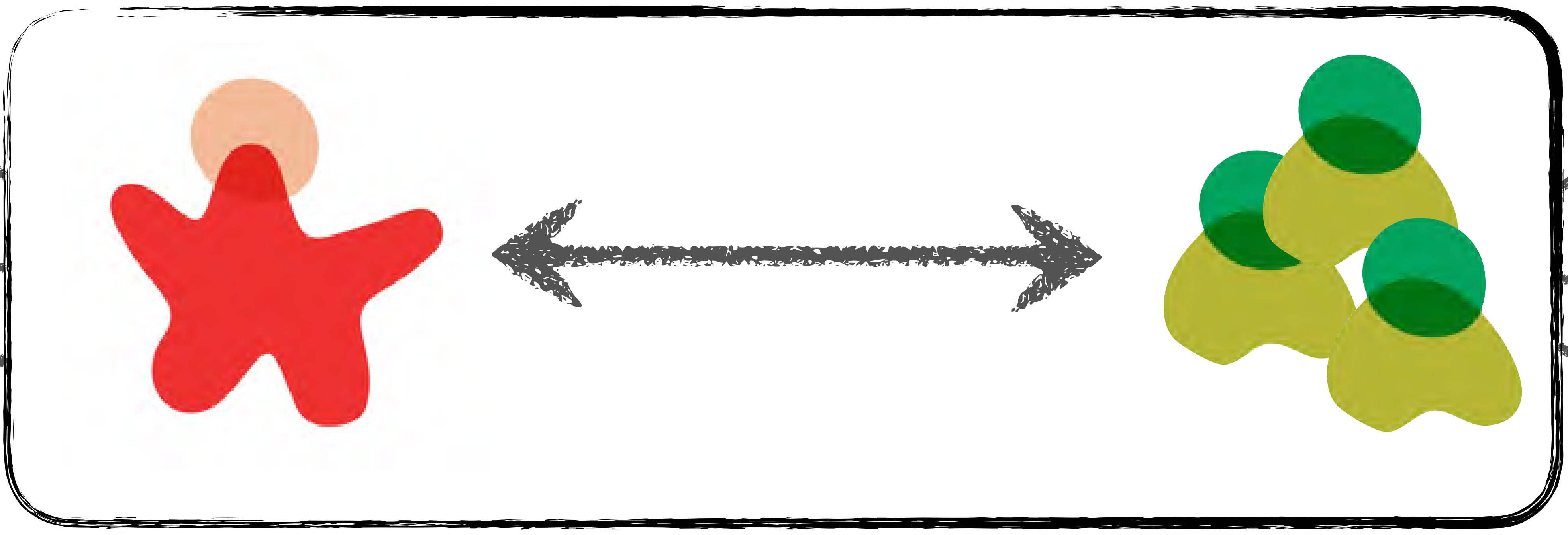
software development

availability
scalability
performance

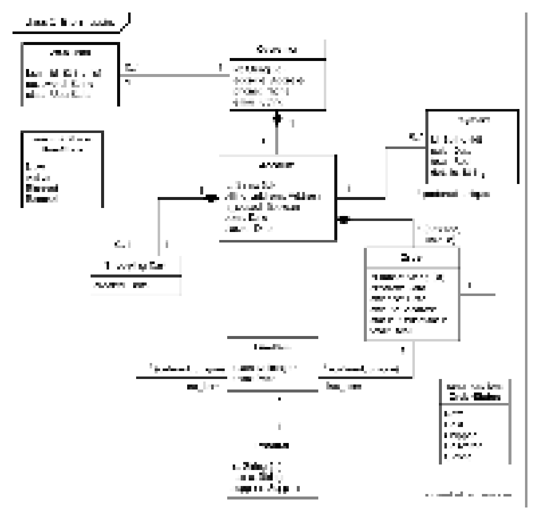


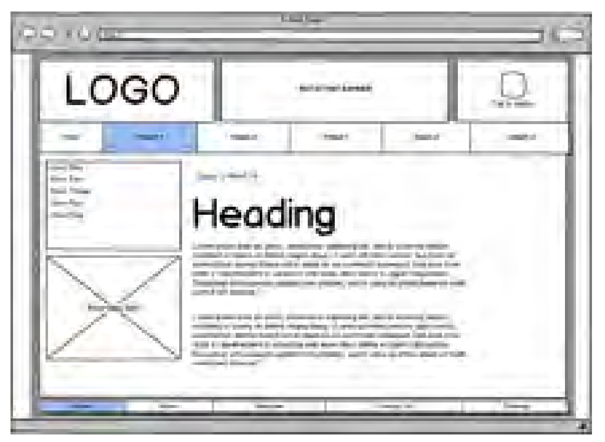


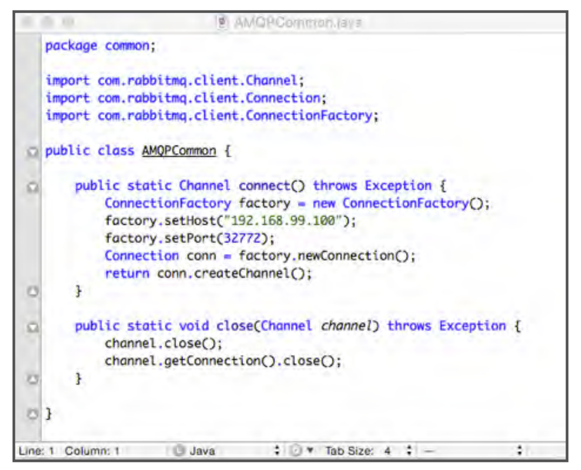
leadership



mentoring





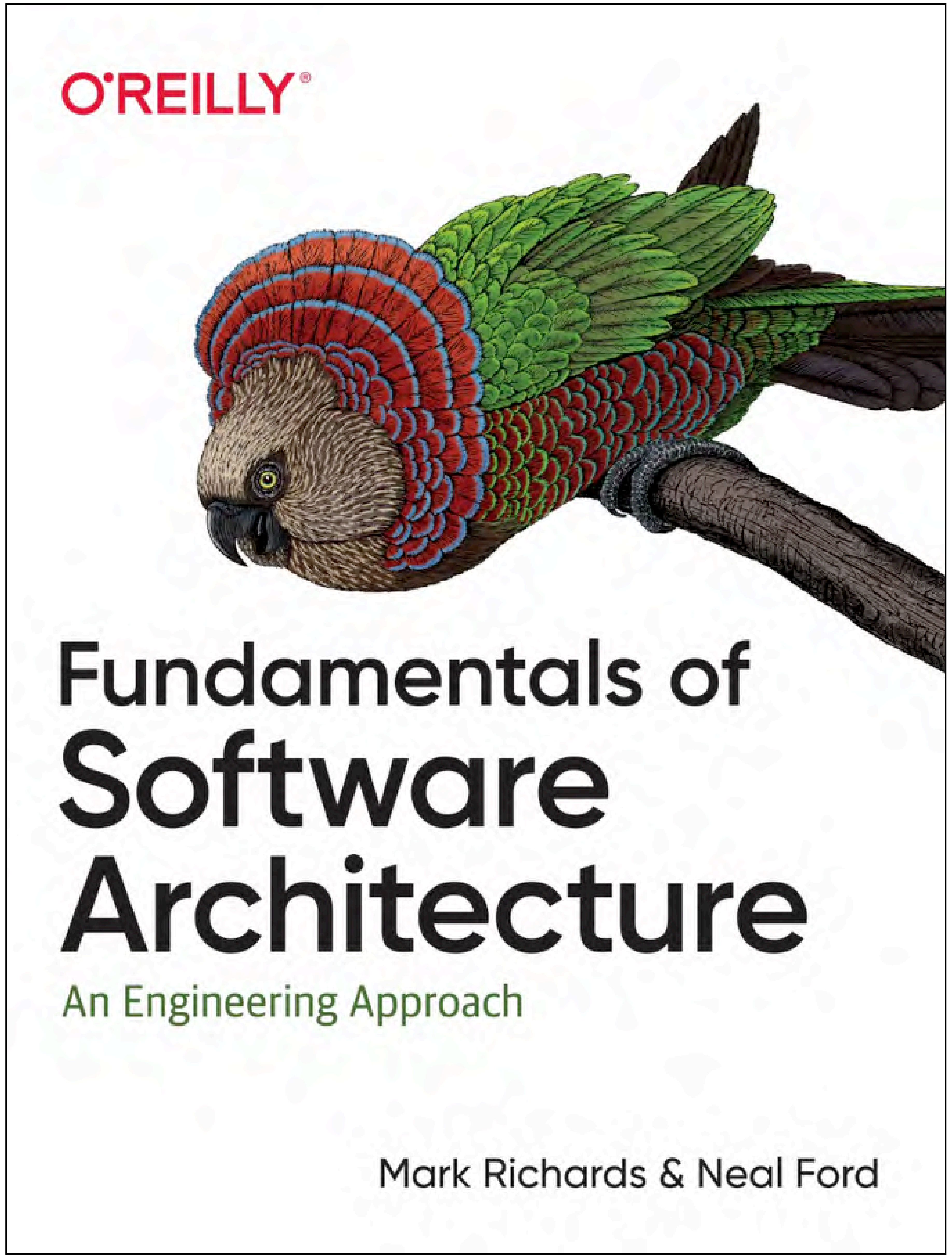


AMQPCommon.java

```
package common;

import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class AMQPCommon {

    public static Channel connect() throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("192.168.99.100");
        factory.setPort(32772);
        Connection conn = factory.newConnection();
        return conn.createChannel();
    }

    public static void close(Channel channel) throws Exception {
        channel.close();
        channel.getConnection().close();
    }

}
```

Line: 1 Column: 1 Java Tab Size: 4

software architecture
software development



O'REILLY

Fundamentals of Software Architecture

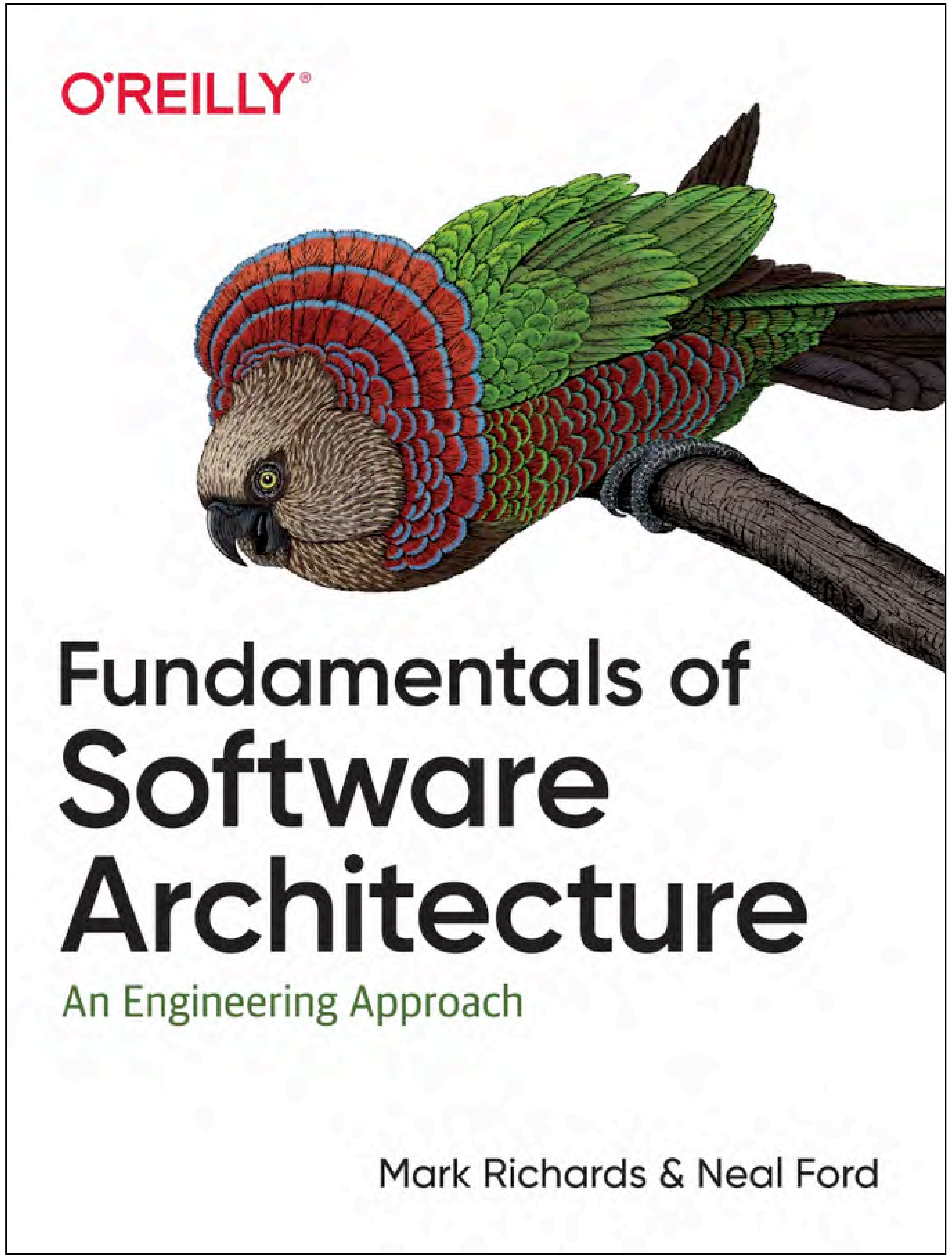
An Engineering Approach

Mark Richards & Neal Ford

"Developers should never take components designed by architects as the last word. Rather, the initial design should be viewed as a first draft, where implementation will reveal more details and refinements."

software architects should adopt and follow best practices in software architecture

REJECTED


O'REILLY

Fundamentals of Software Architecture

An Engineering Approach

Mark Richards & Neal Ford

First Law of Software Architecture

"Everything in software architecture is a trade-off"

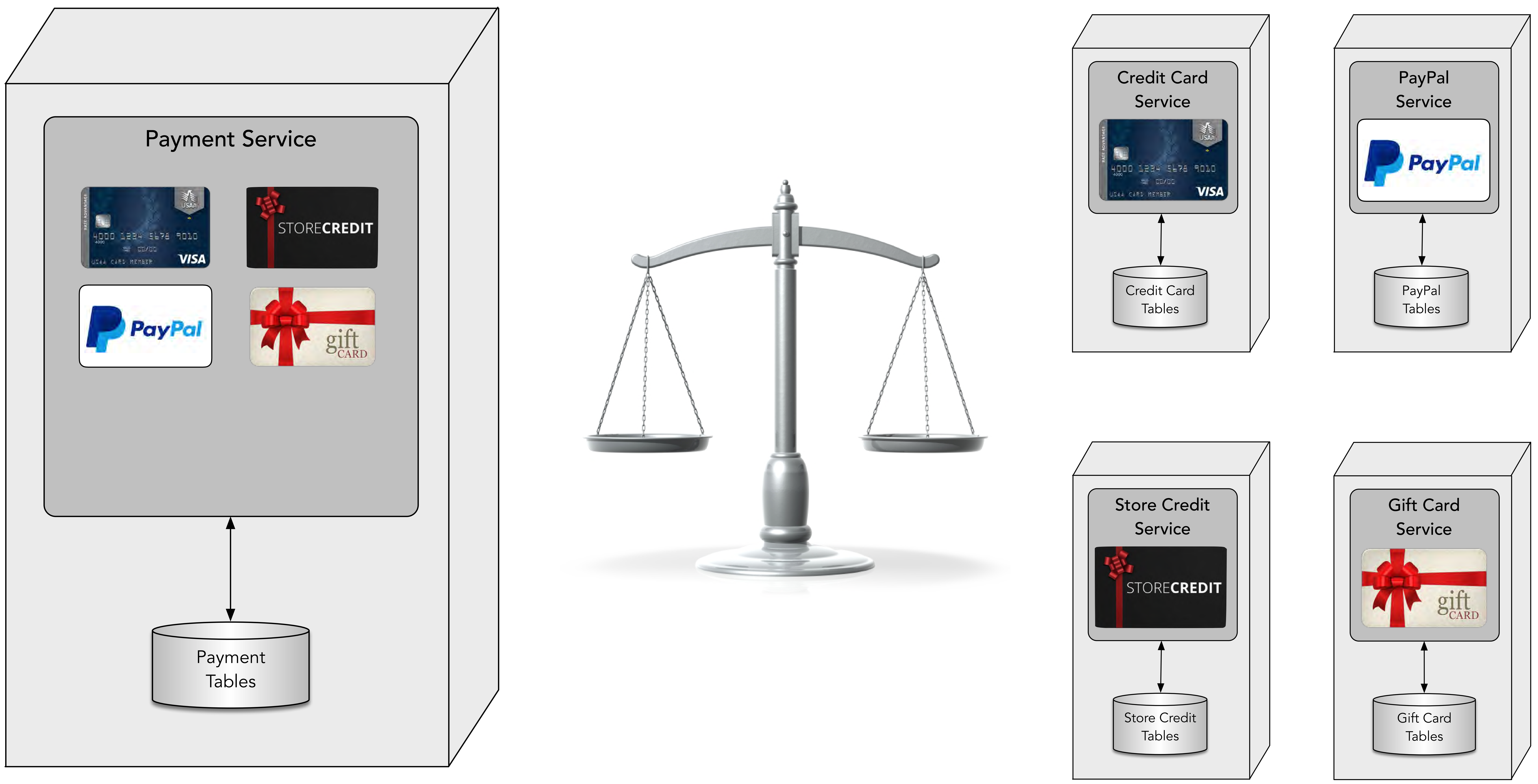
there are no best practices!

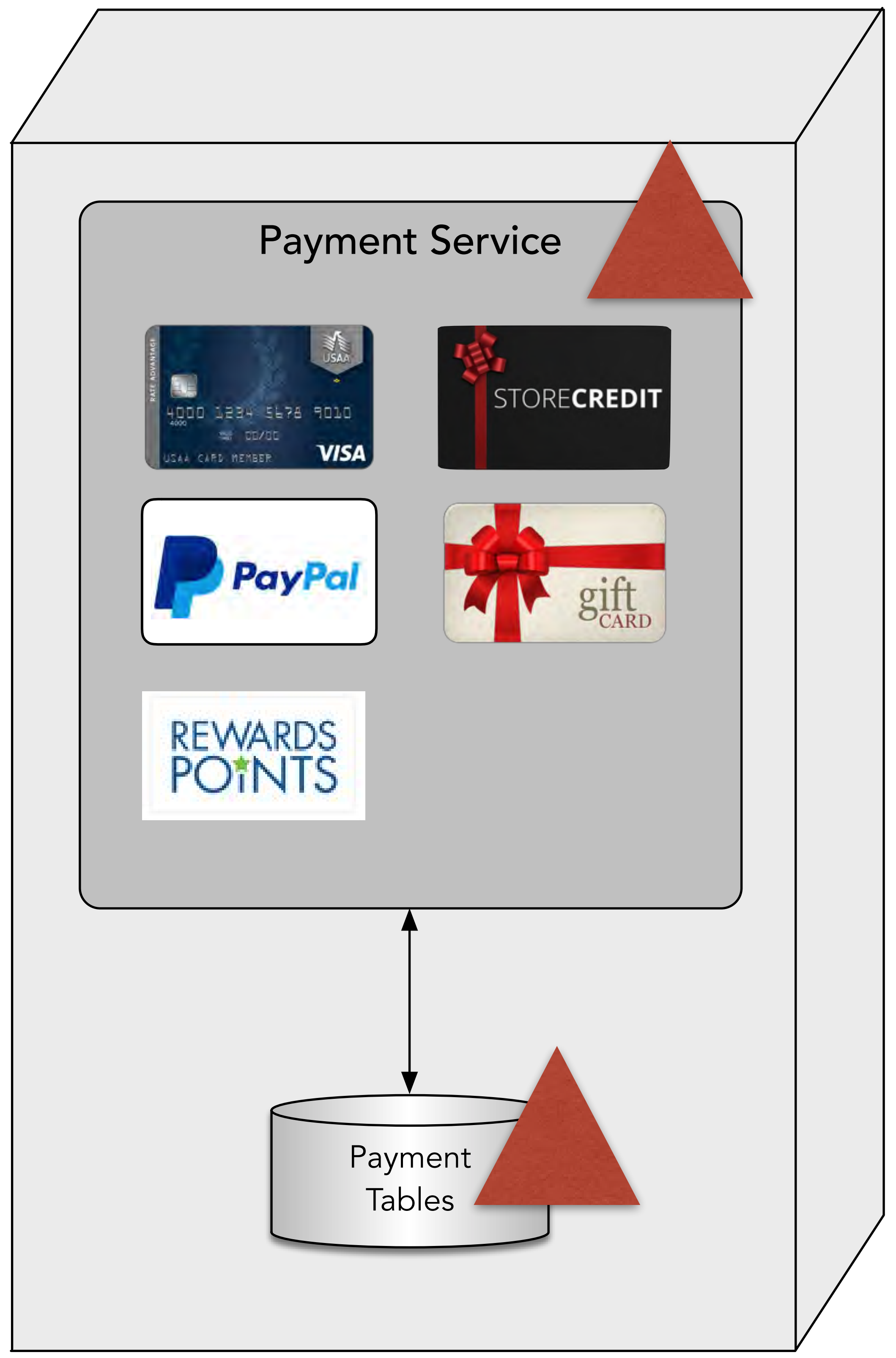
USAA
4000 1234 5678 9010
USAA CARD MEMBER
VISA

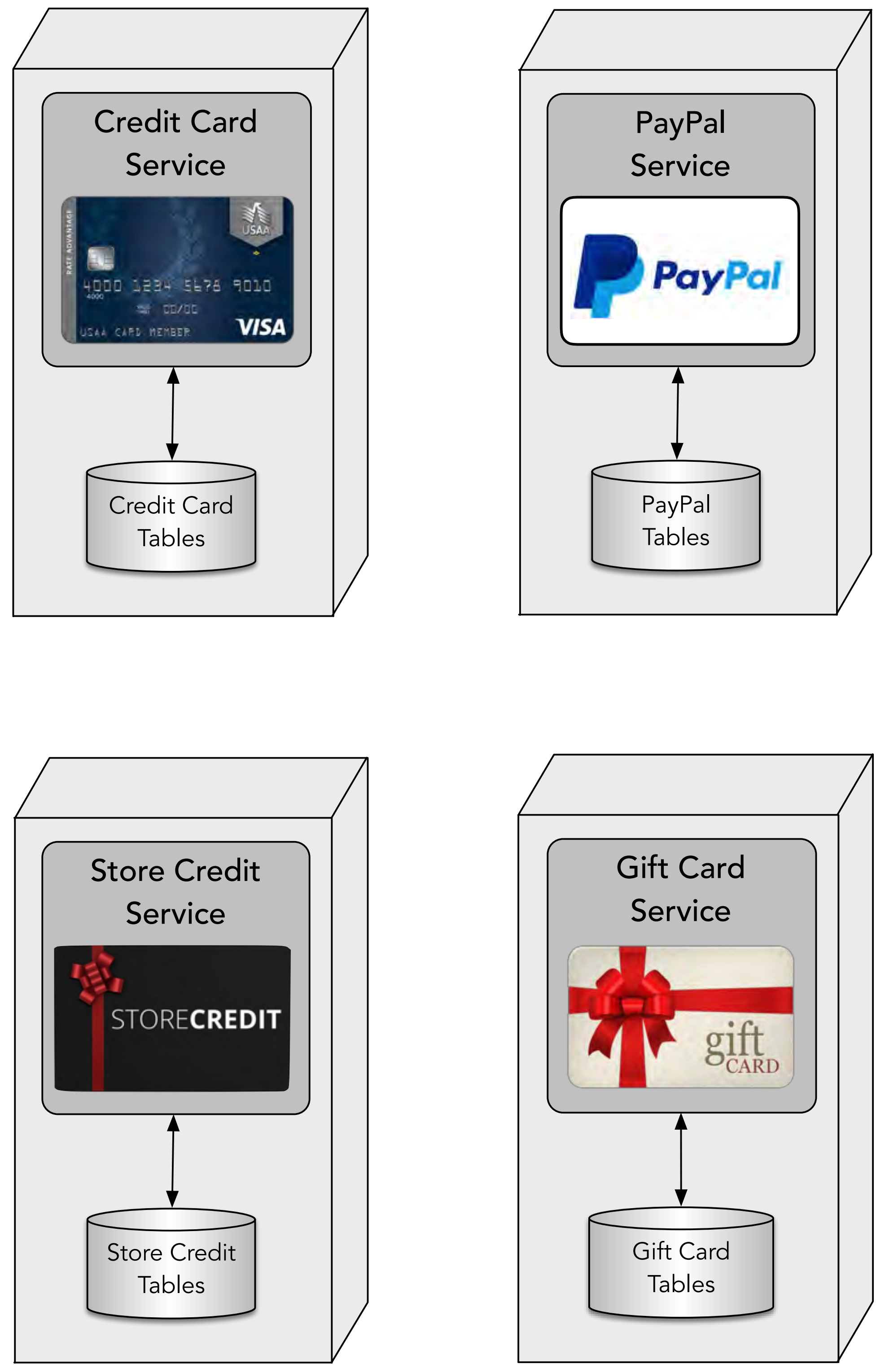
PayPal

STORECREDIT

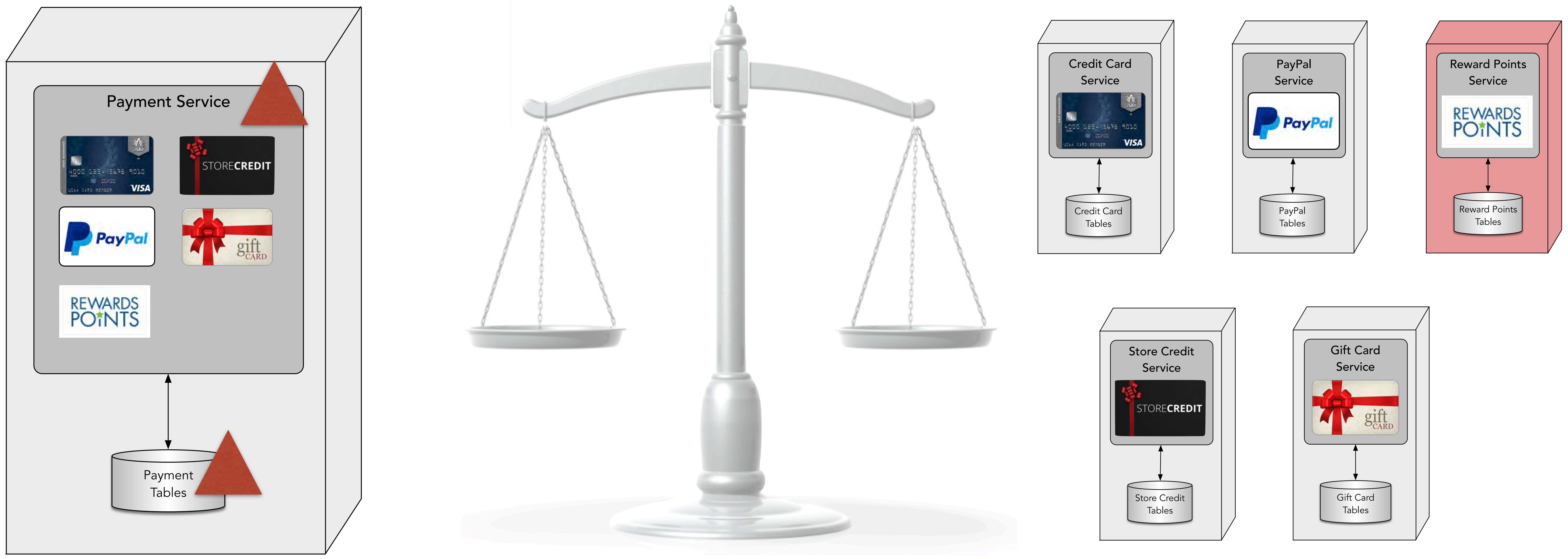
gift CARD


Payment Service
STORECREDIT
PayPal
gift CARD
VISA
Payment Tables
Credit Card Service
Credit Card Tables
PayPal Service
PayPal Tables
Store Credit Service
Store Credit Tables
Gift Card Service
Gift Card Tables

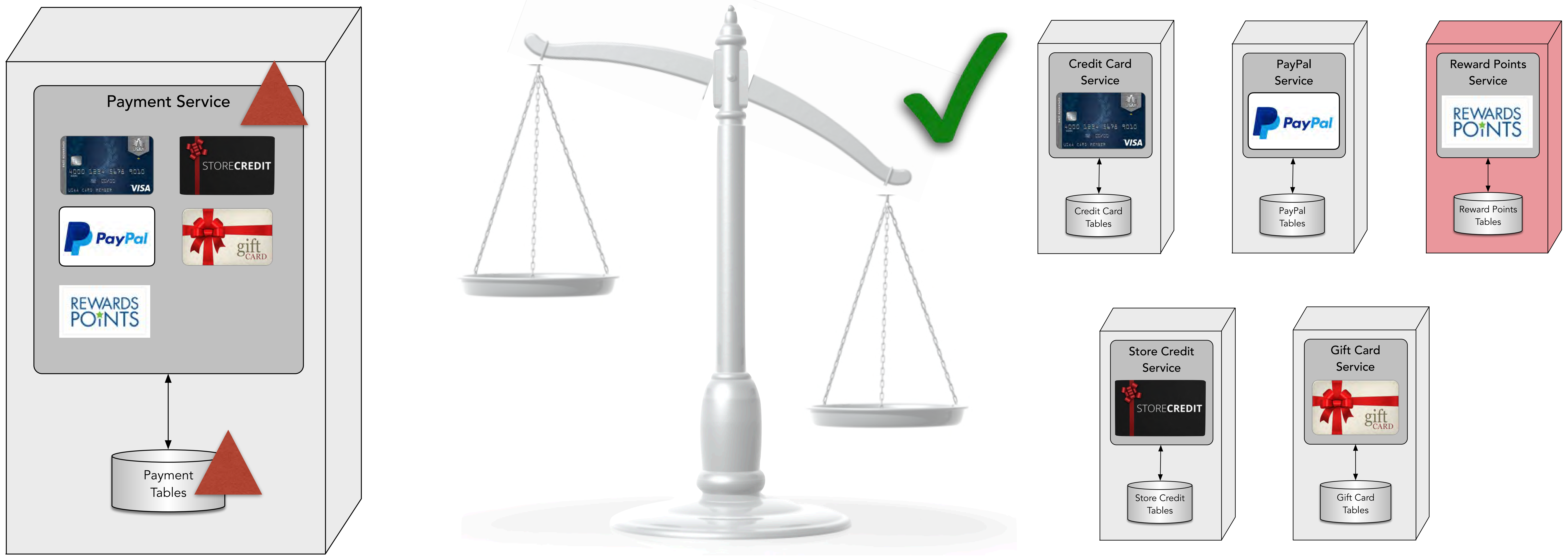




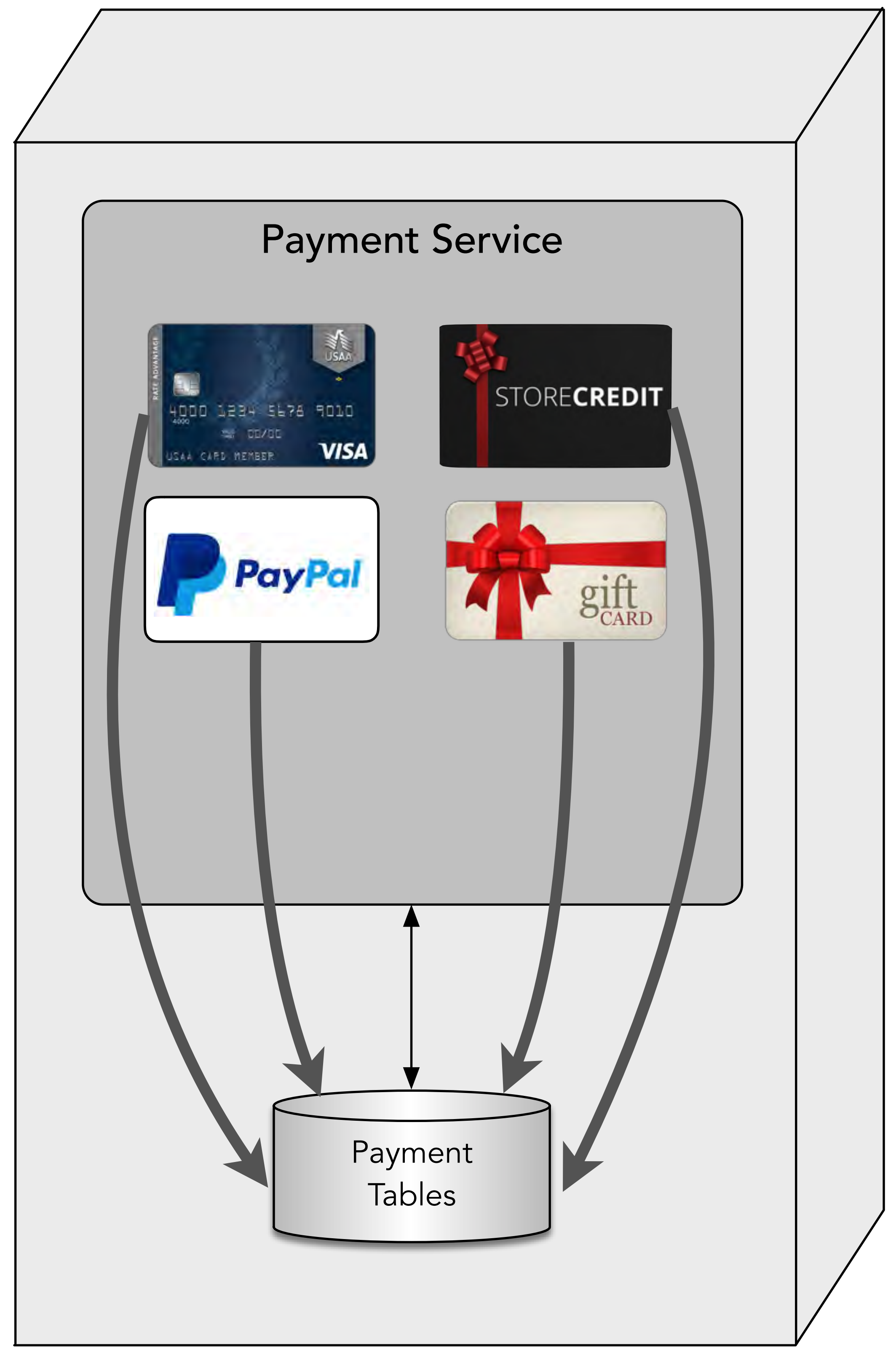
add a new feature to collect, maintain, and redeem reward points

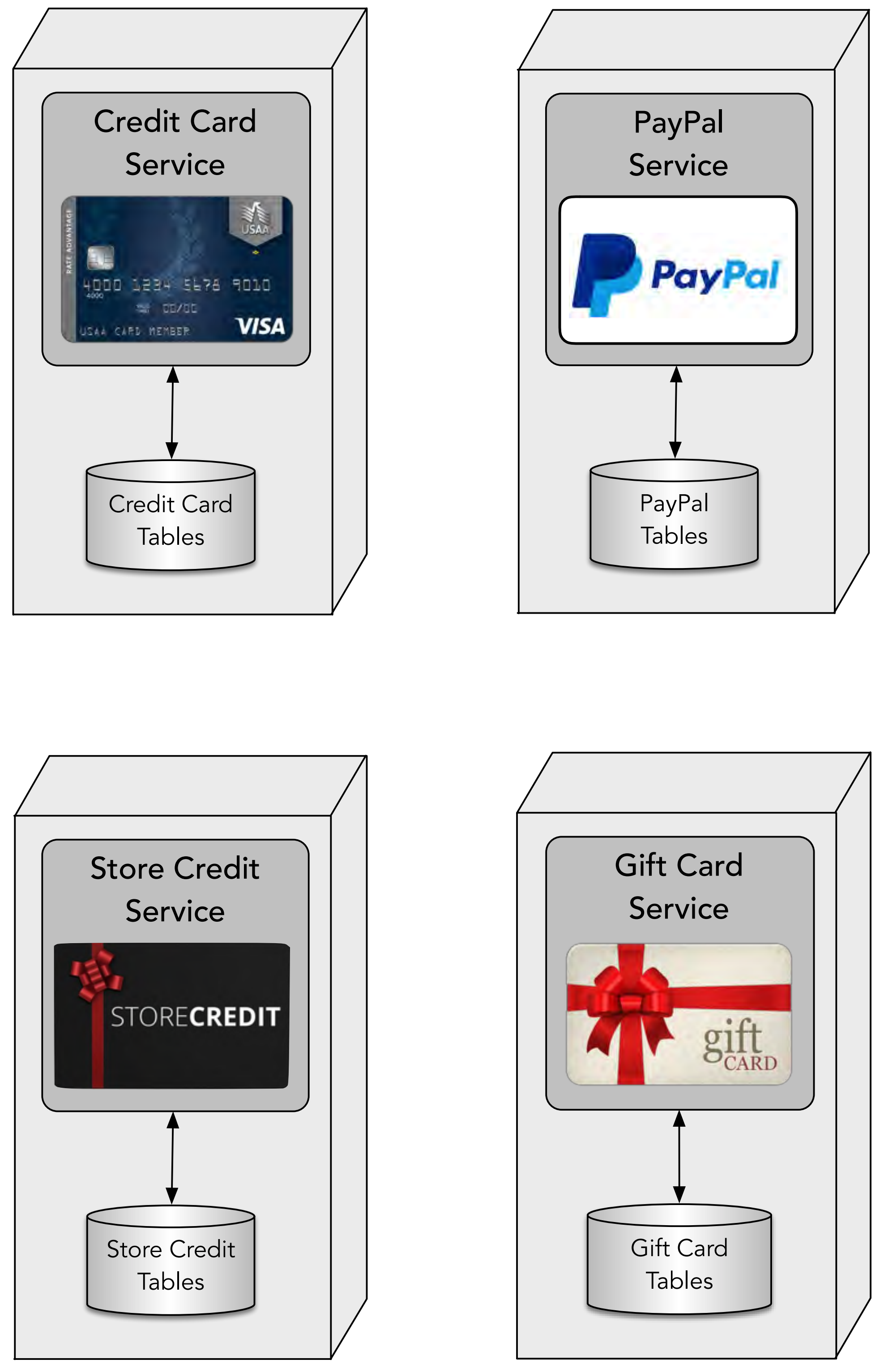


add a new feature to collect, maintain, and redeem reward points

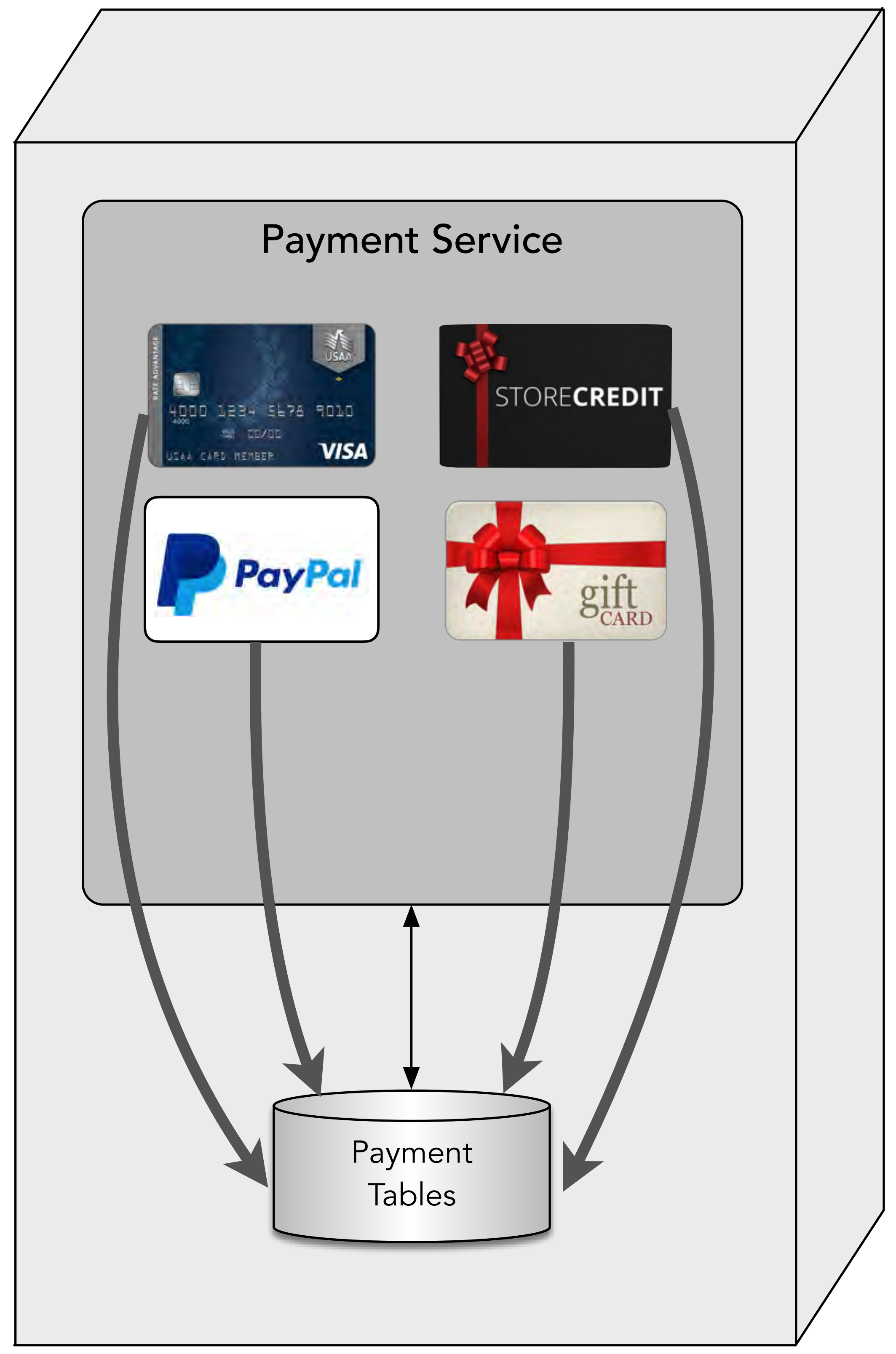


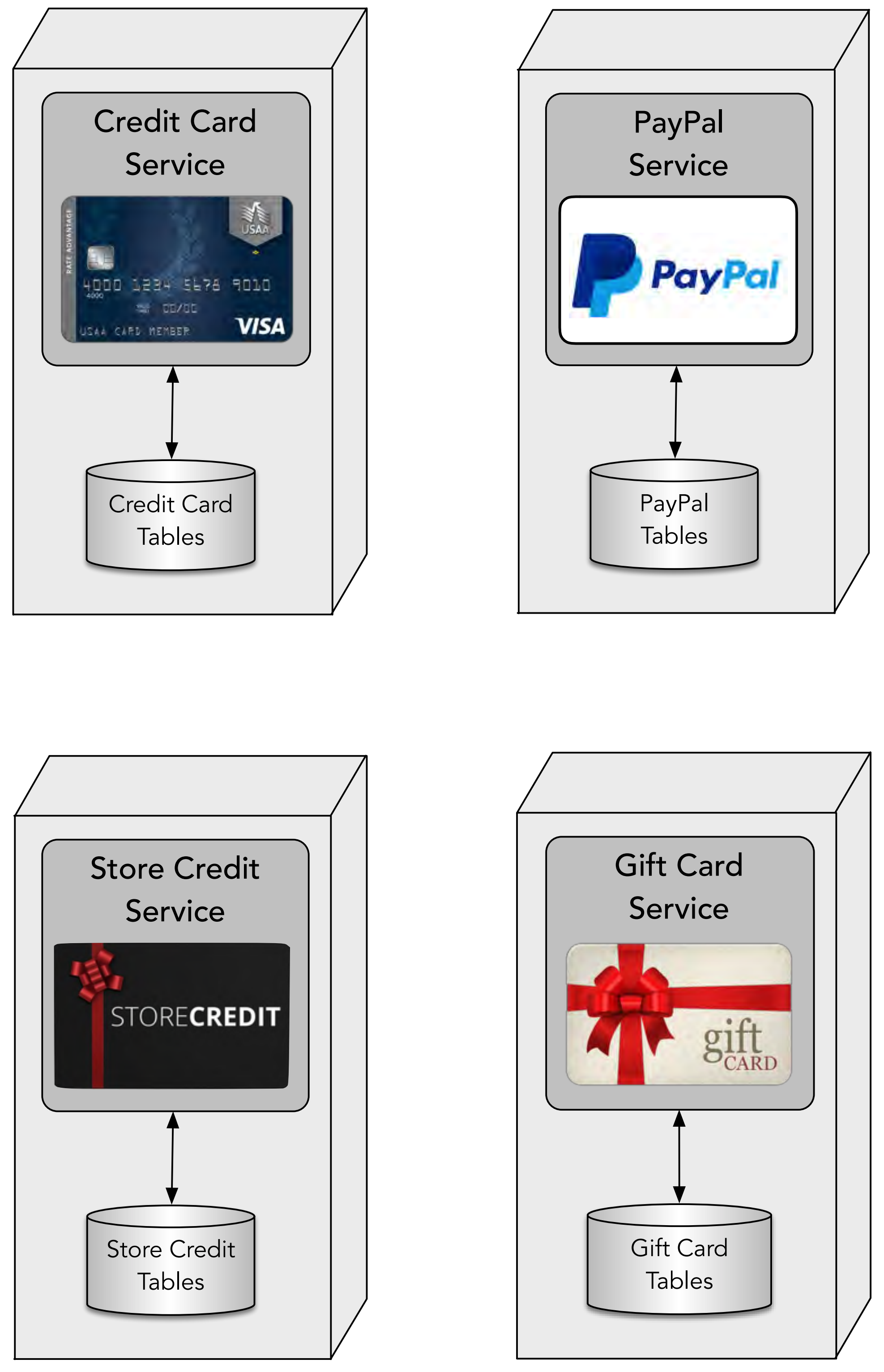
add a new feature to collect, maintain, and redeem reward points



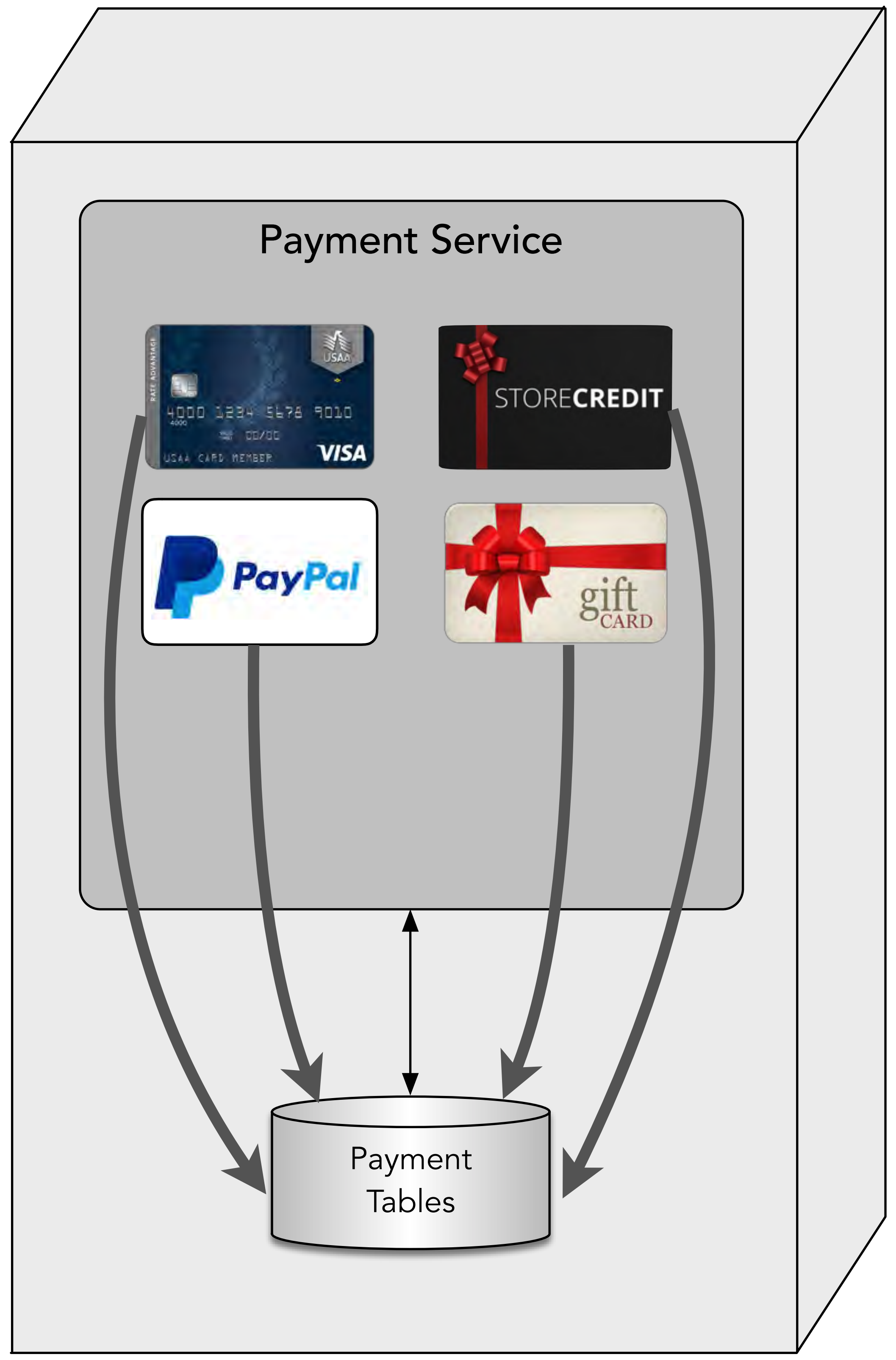


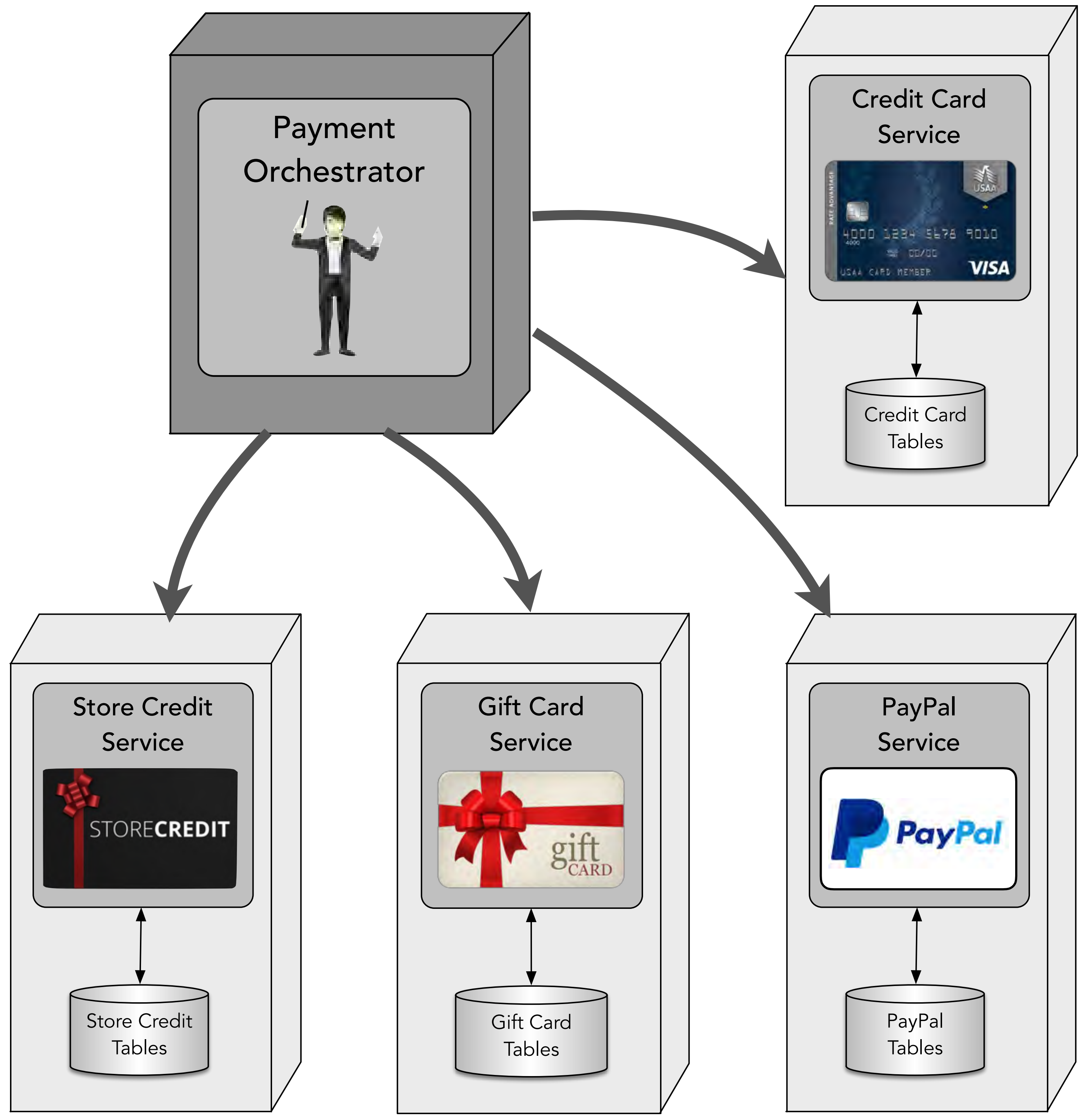
add the ability to apply multiple payment types to pay for an order



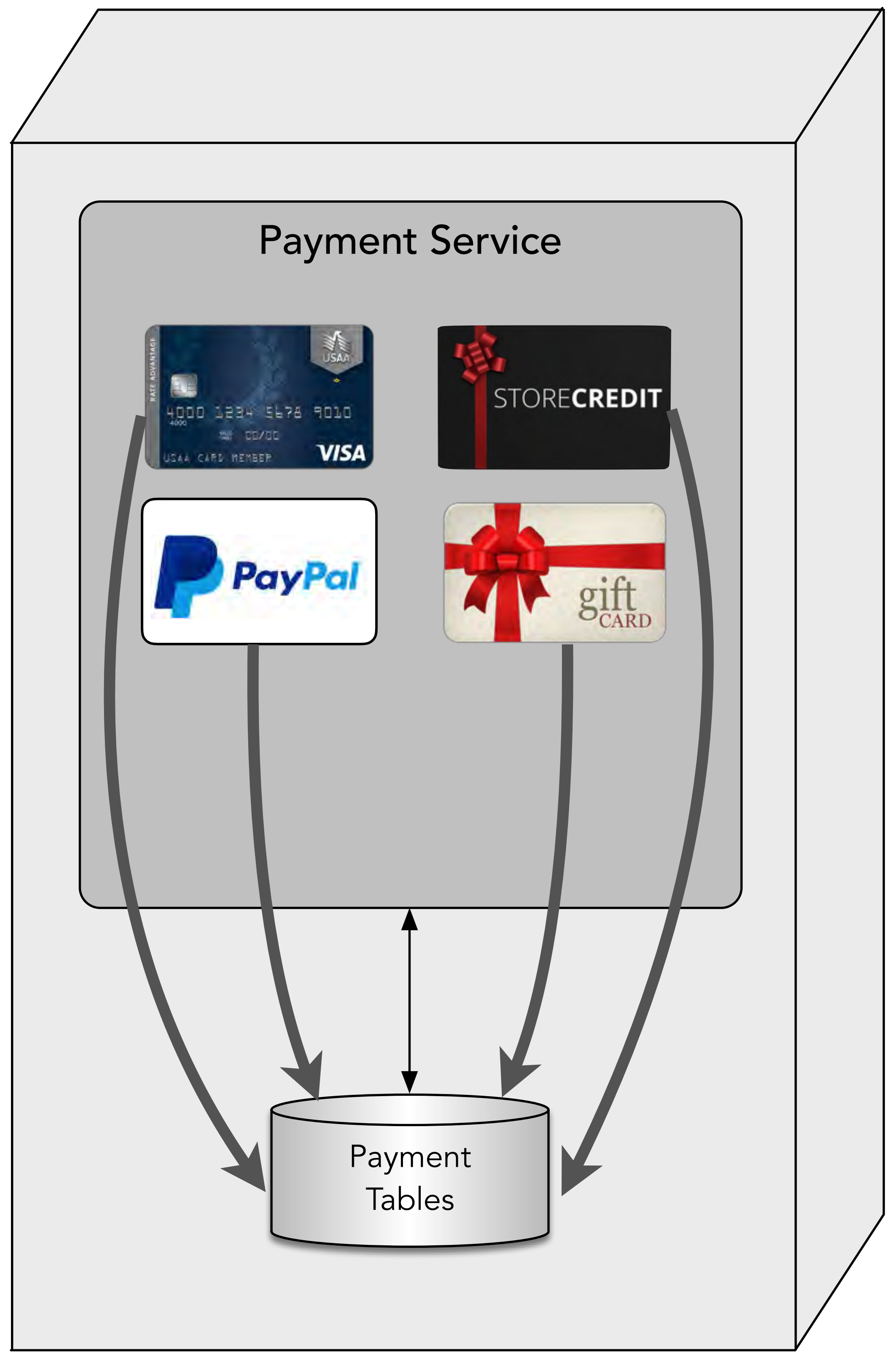


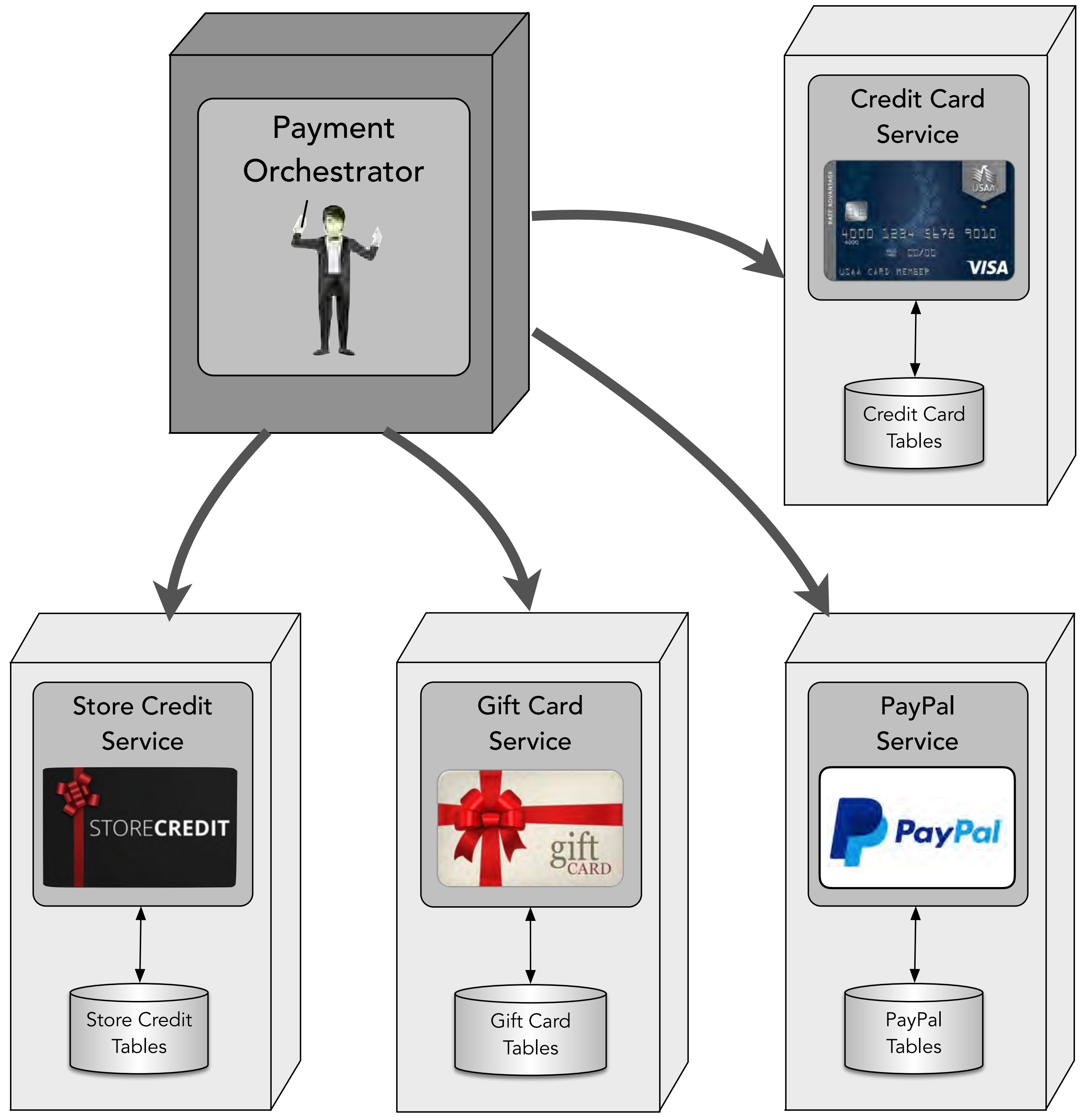
add the ability to apply multiple payment types to pay for an order



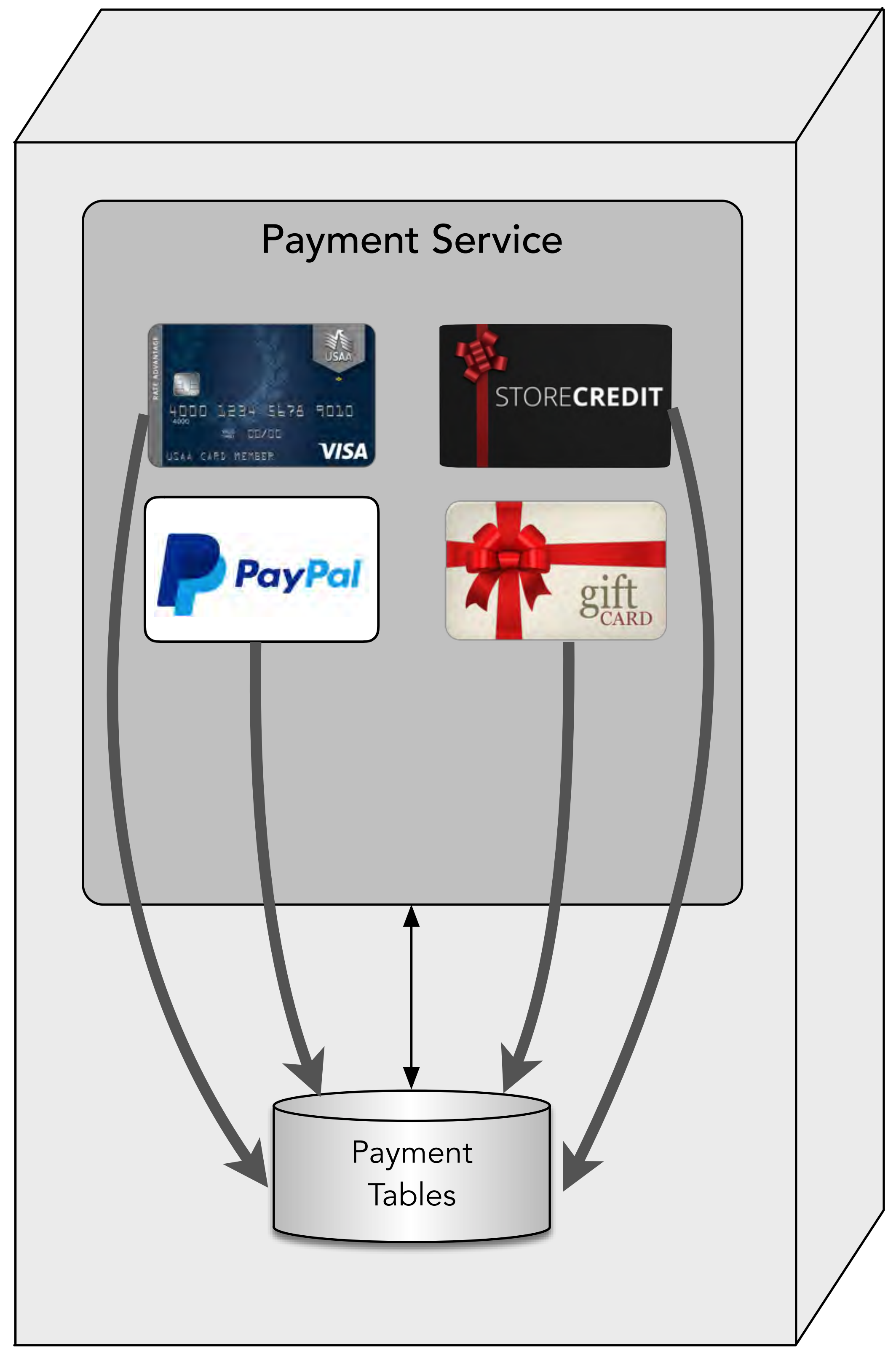


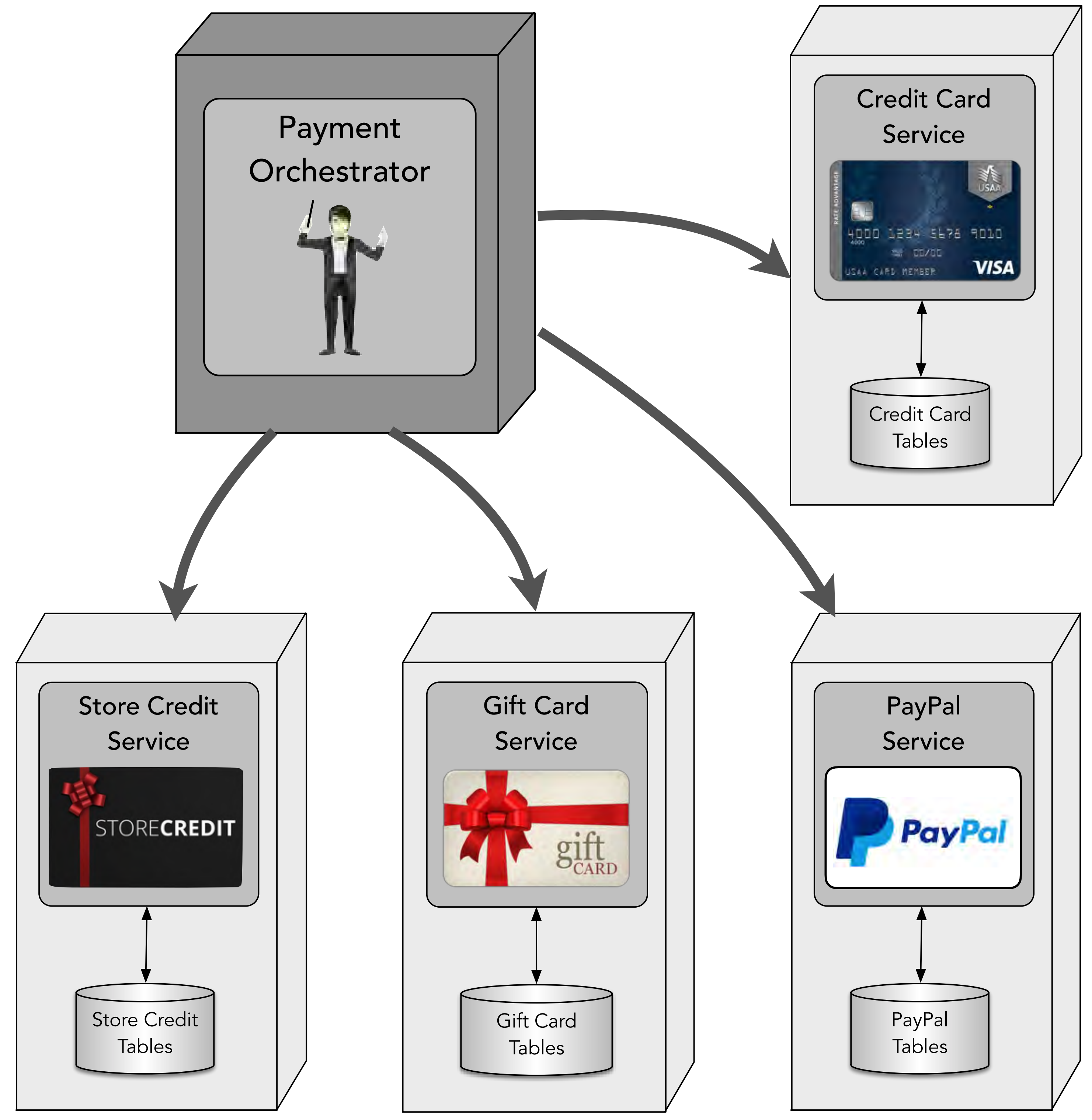
add the ability to apply multiple payment types to pay for an order





add the ability to apply multiple payment types to pay for an order





add the ability to apply multiple payment types to pay for an order

there are no best practices in software architecture - only trade-offs

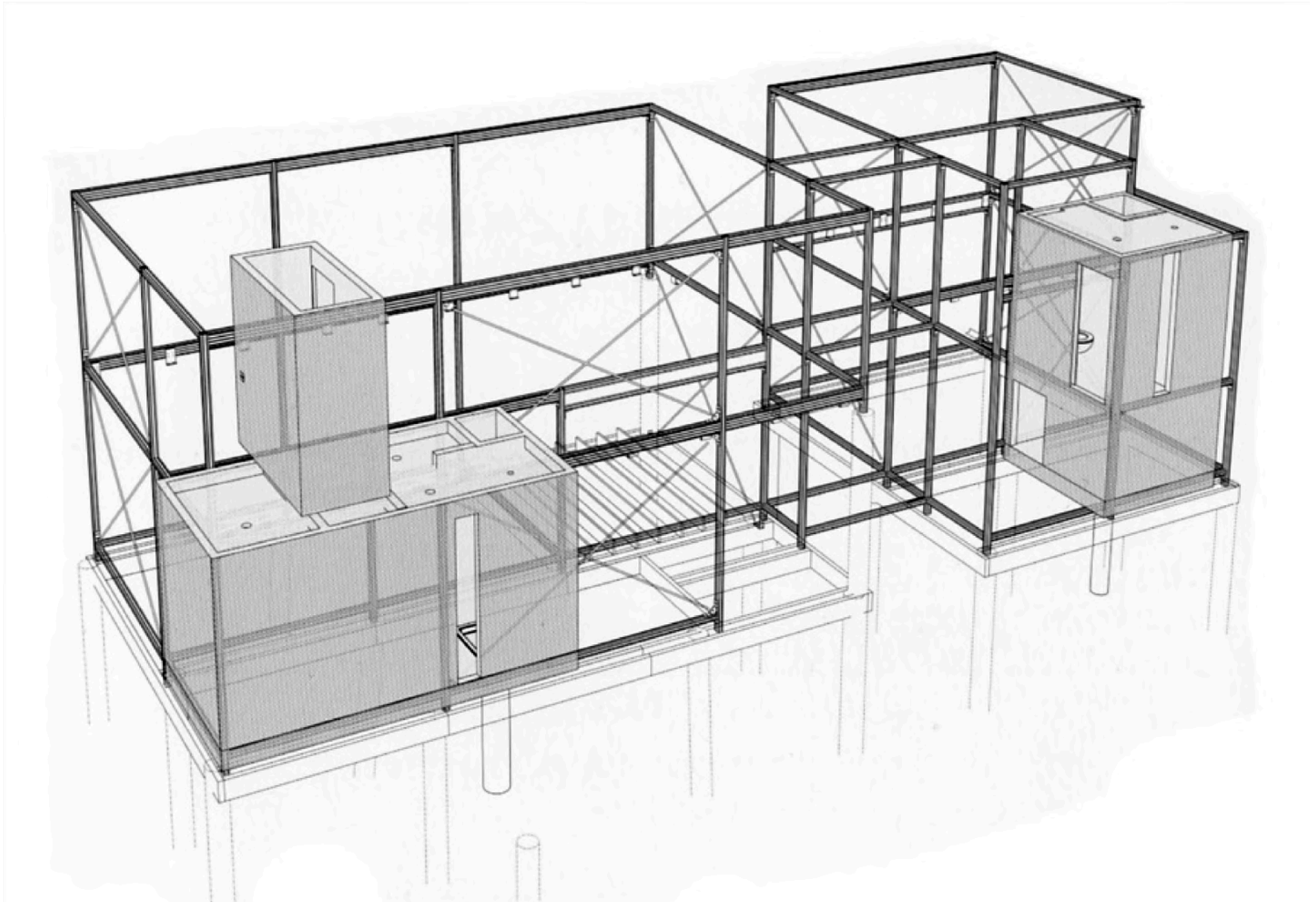
software architecture is the foundational structure of a system and therefore should not undergo frequent change

REJECTED

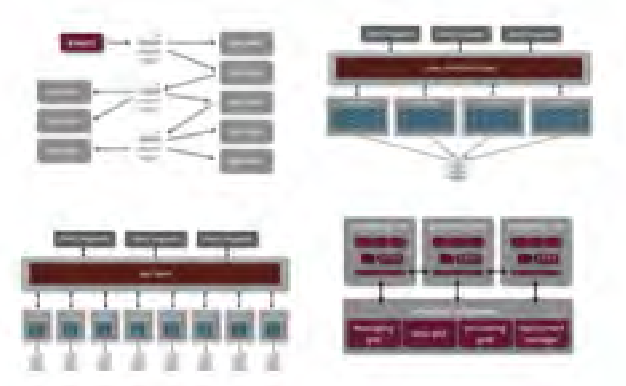
software architecture is the stuff that's hard to change later

REJECTED

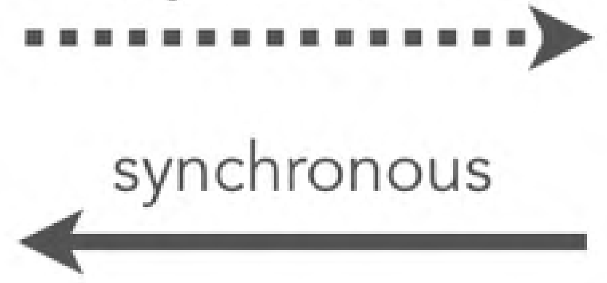
continually refactoring the architecture
means you don't know what you are doing



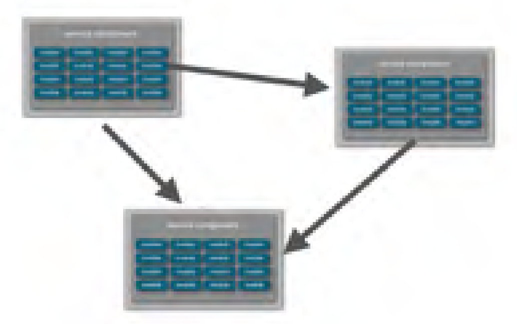
architecture



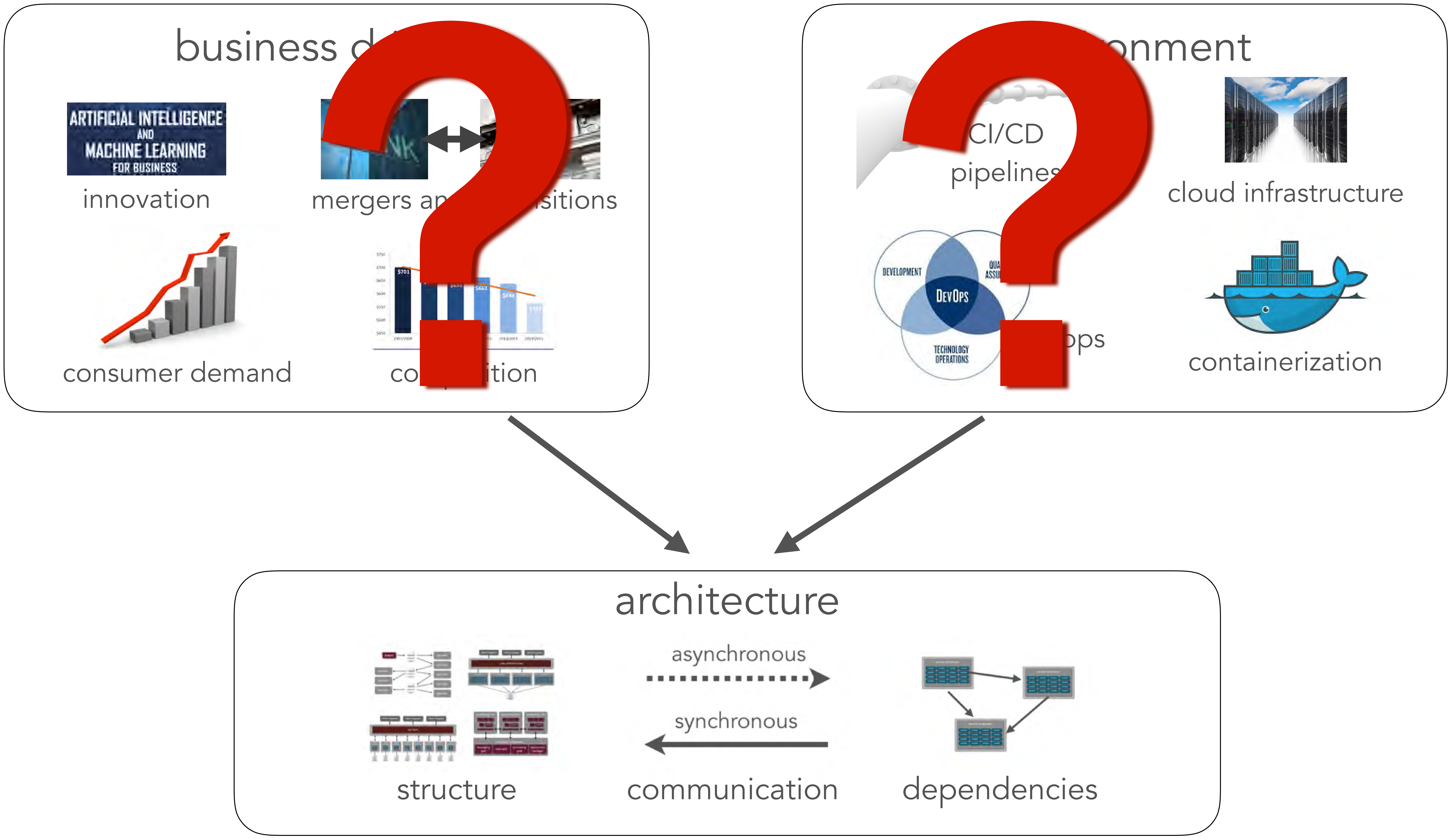
structure

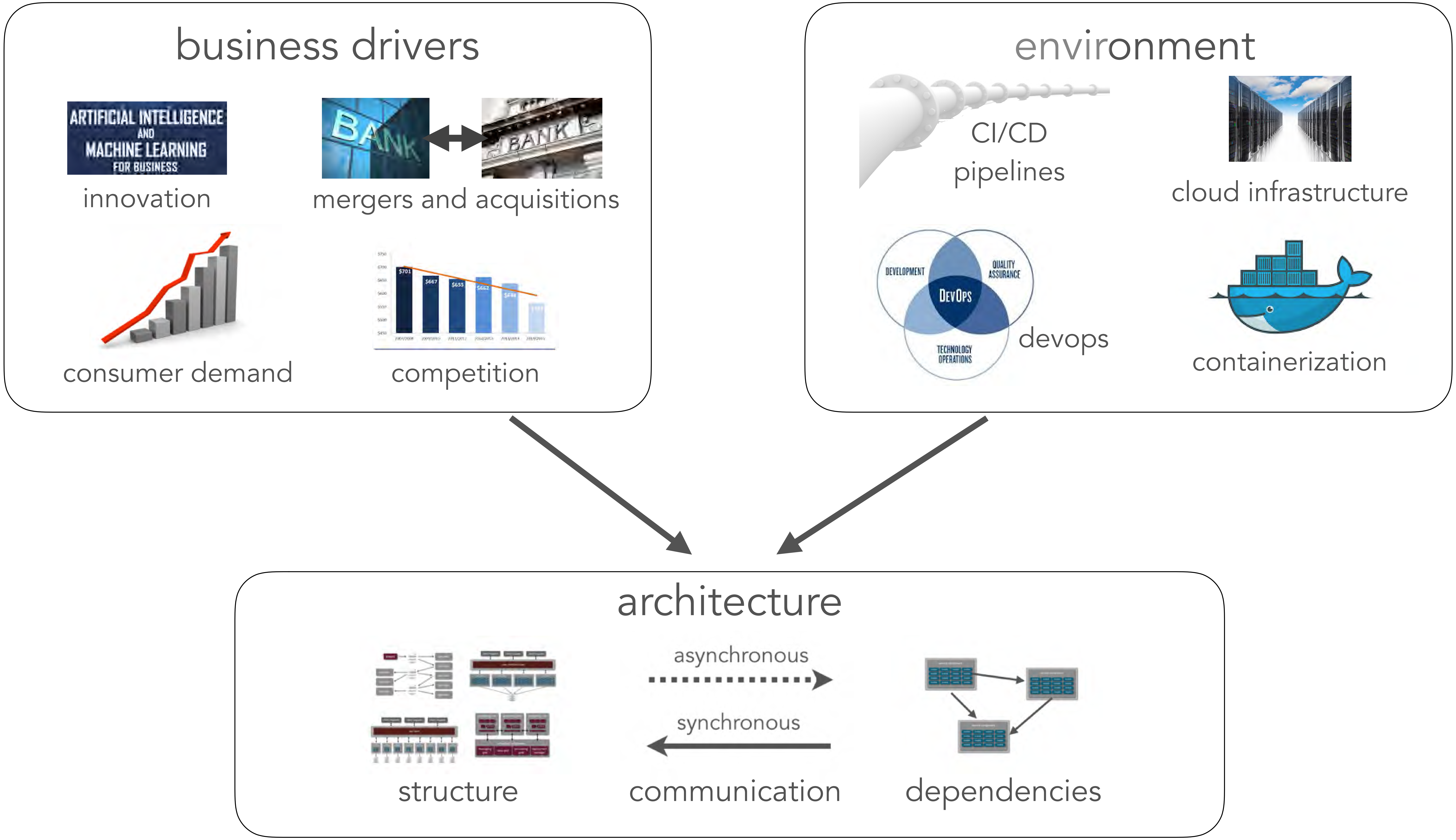


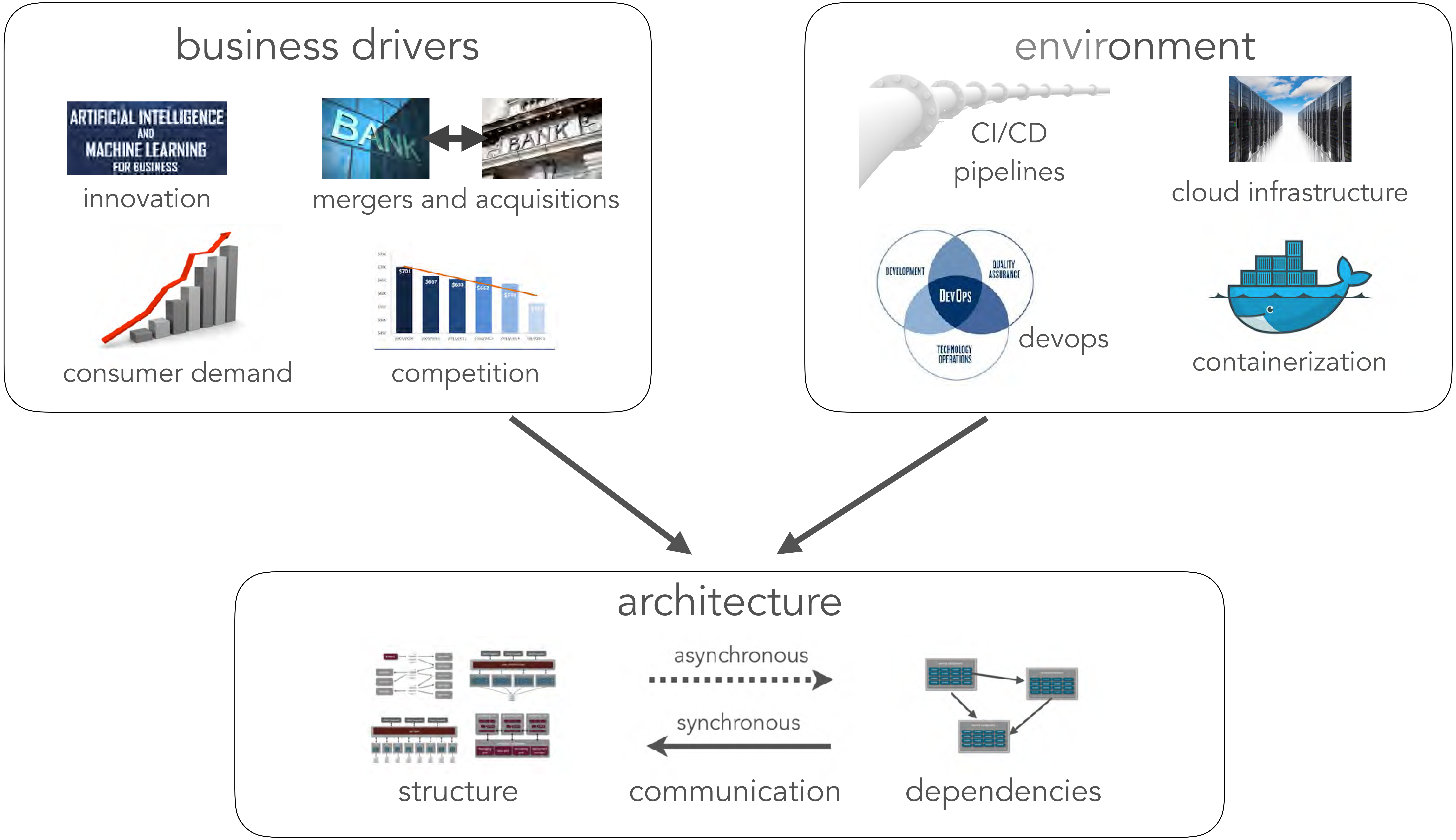
communication



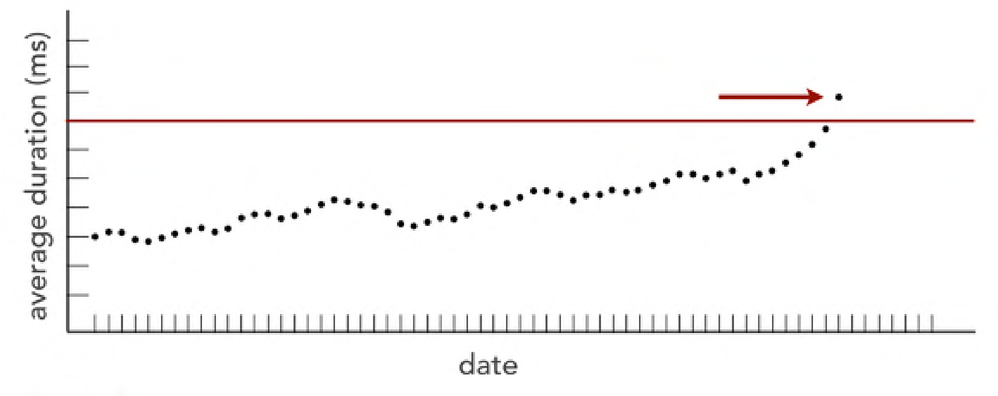
dependencies


business d
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR BUSINESS
innovation
mergers an
sitions
consumer demand
co
ition
onment
CI/CD
pipelines
cloud infrastructure
DEVELOPMENT
DEVOPS
TECHNOLOGY OPERATIONS
ops
containerization
architecture
asynchronous
synchronous
structure
communication
dependencies

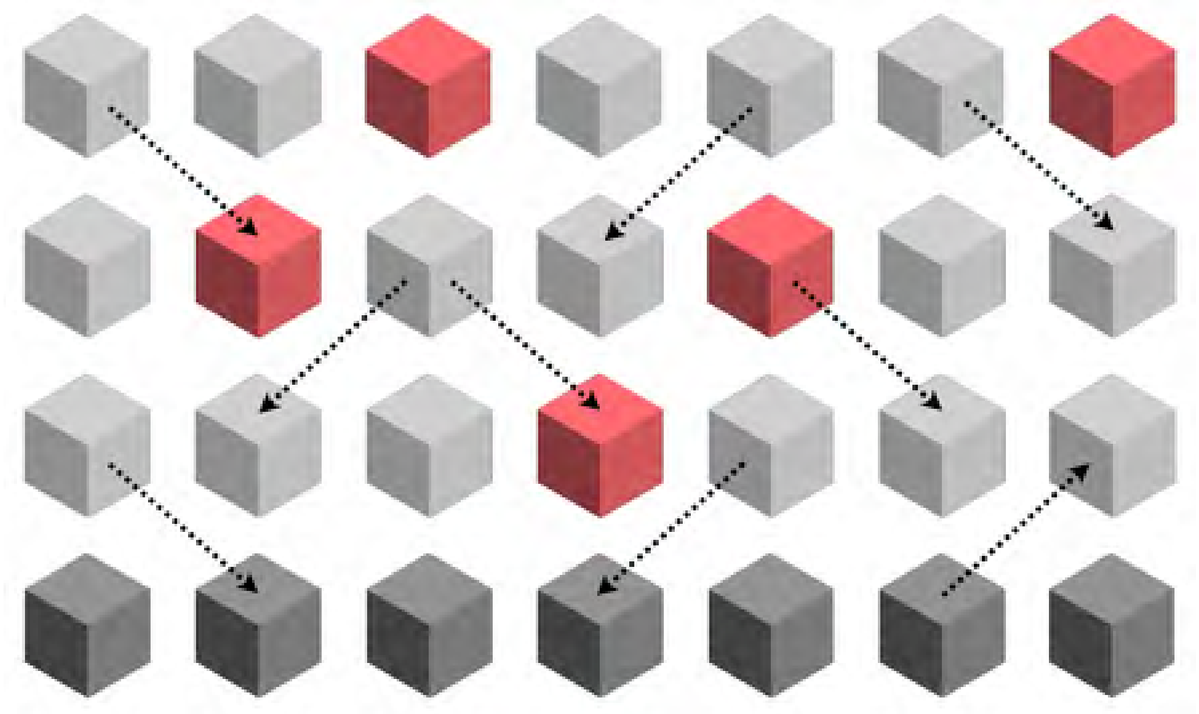

business drivers
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR BUSINESS
innovation
BANK
BANK
mergers and acquisitions
consumer demand
$701
$667
$655
$662
competition
environment
CI/CD pipelines
cloud infrastructure
DEVELOPMENT
QUALITY ASSURANCE
DEVOPS
TECHNOLOGY OPERATIONS
devops
containerization
architecture
structure
asynchronous
synchronous
communication
dependencies


business drivers
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR BUSINESS
innovation
BANK
BANK
mergers and acquisitions
consumer demand
$701
$667
$655
$662
competition
environment
CI/CD pipelines
cloud infrastructure
DEVELOPMENT
QUALITY ASSURANCE
DEVOPS
TECHNOLOGY OPERATIONS
devops
containerization
architecture
structure
asynchronous
synchronous
communication
dependencies

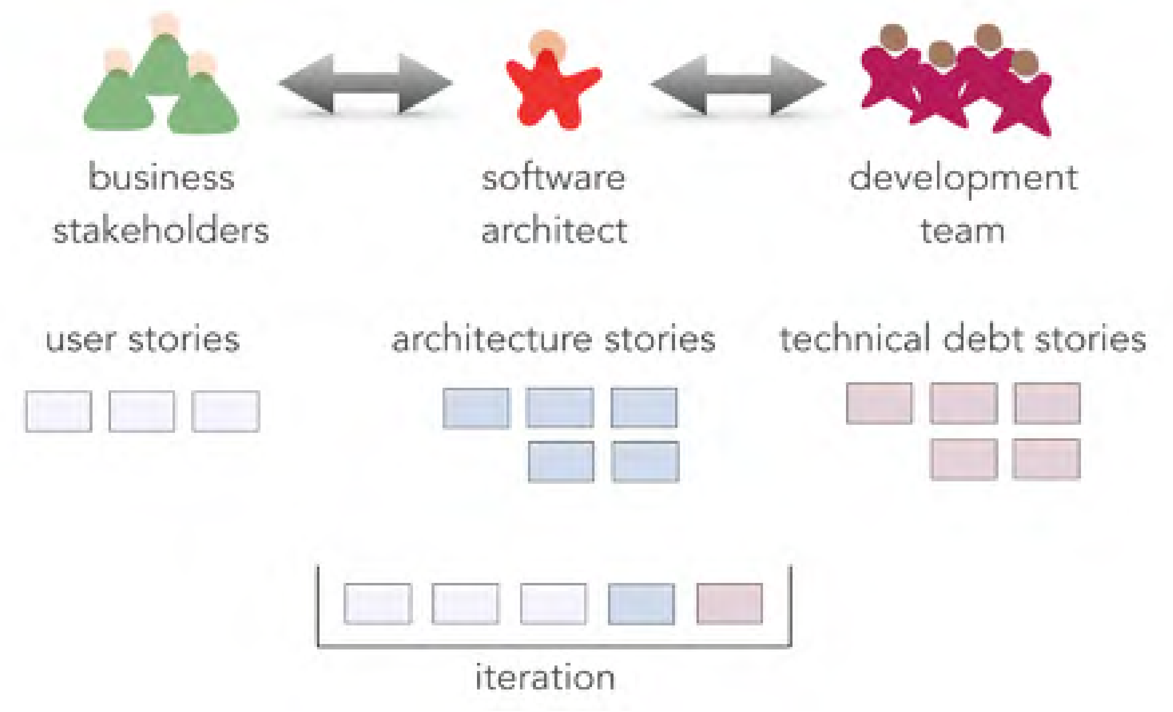
how do software architects handle all of this change?



detect change

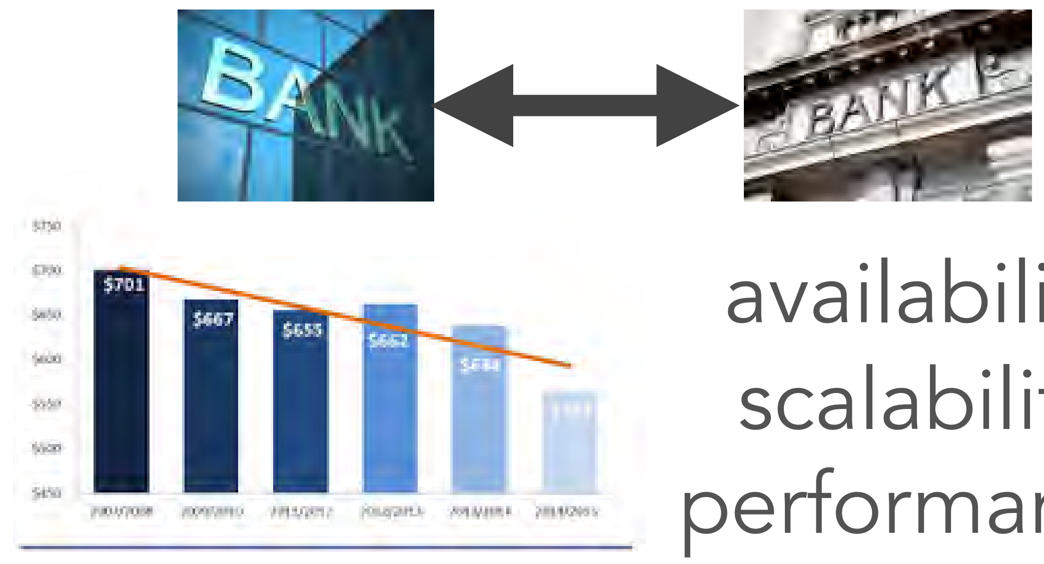


plan for change



facilitate change

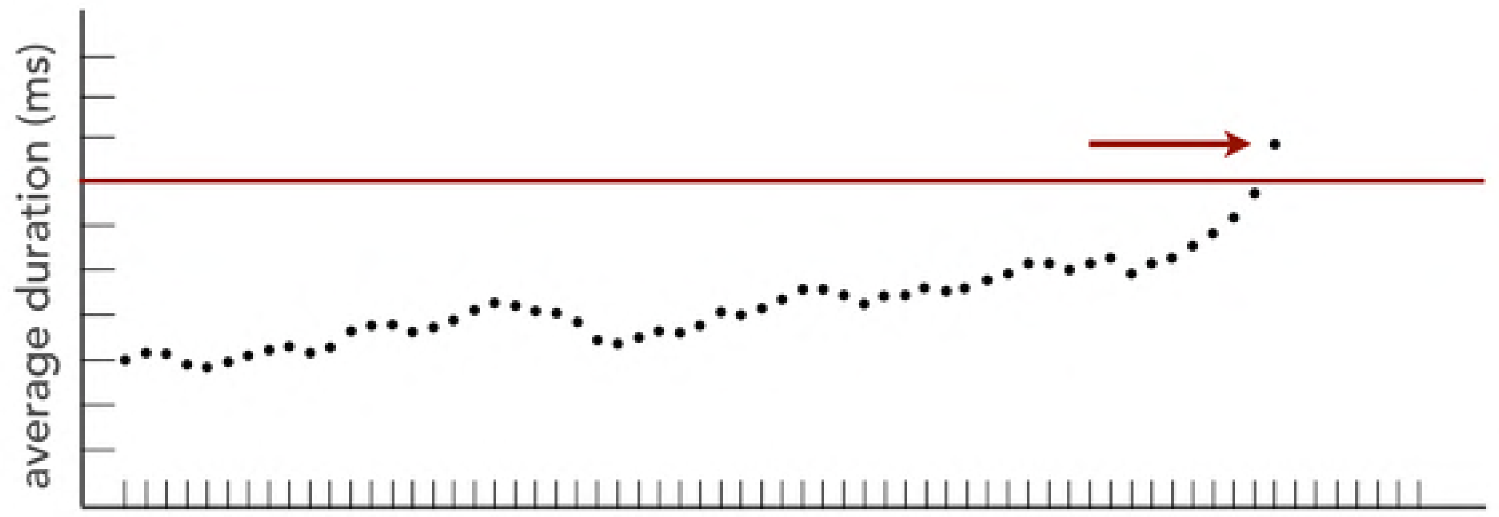
detect change



availability
scalability
performance

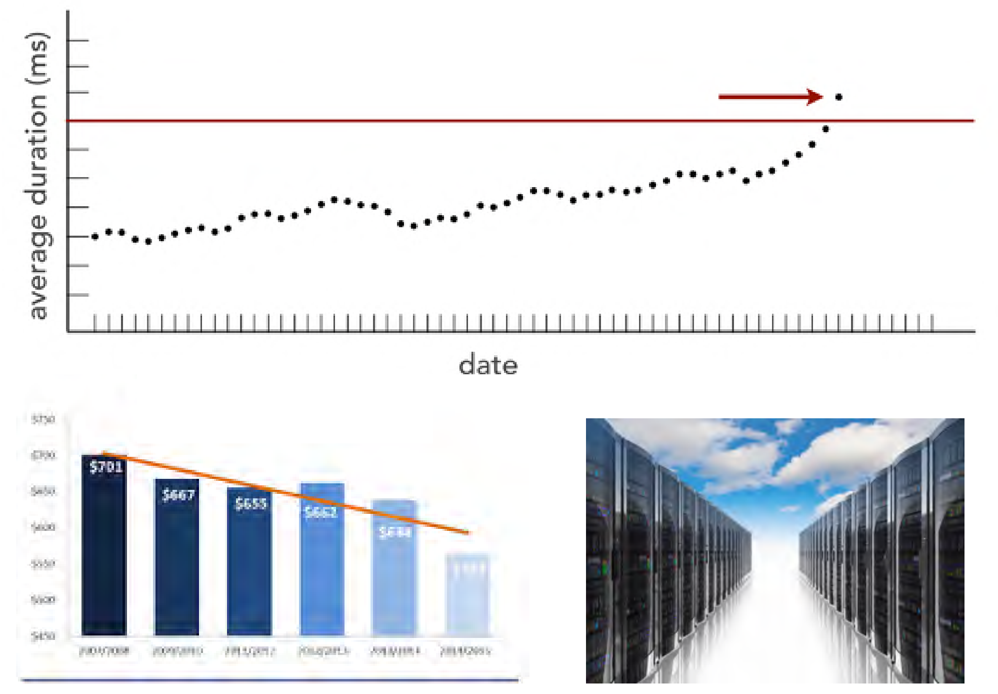
collaborating with
business stakeholders

collaborating with
operations stakeholders

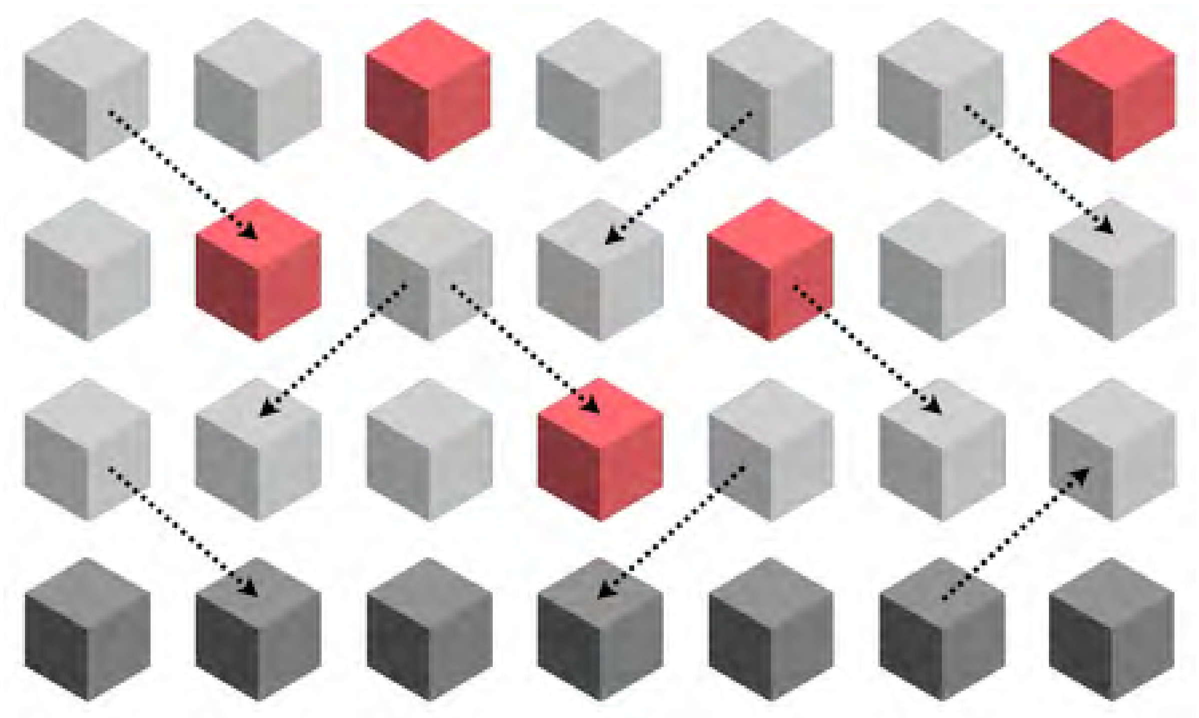


continually analyzing
architecture characteristics

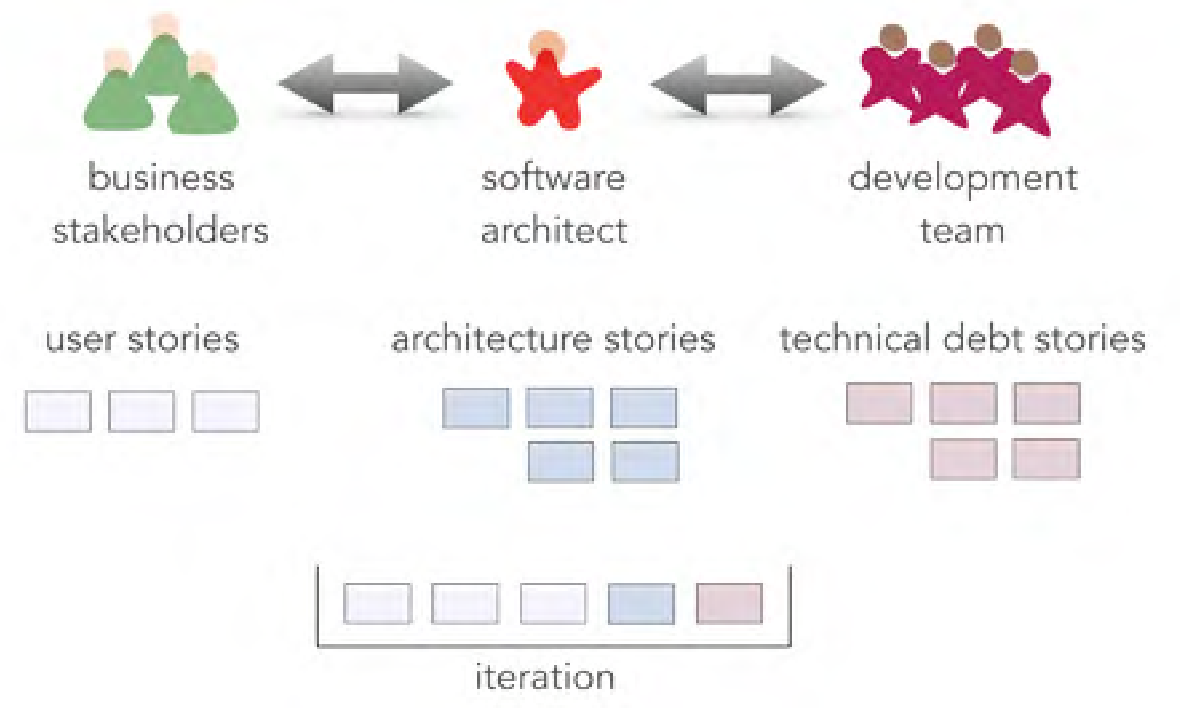
how do software architects handle all of this change?



detect change

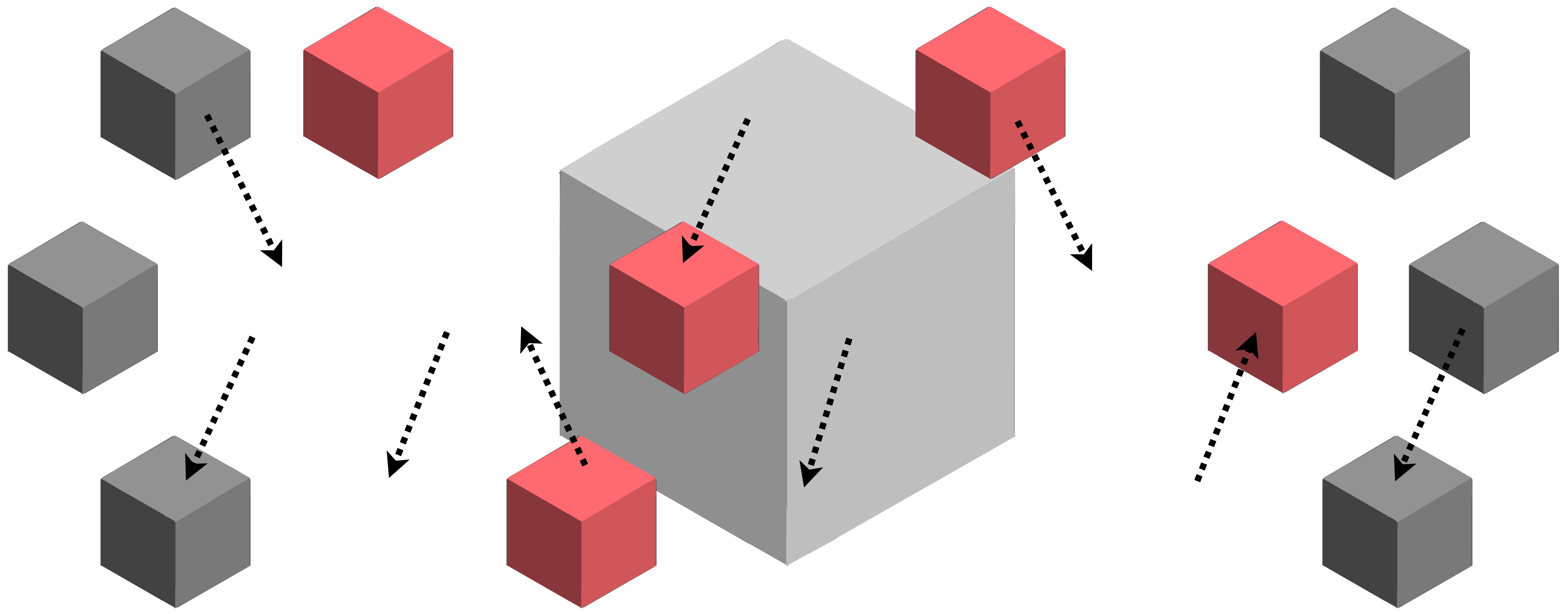


plan for change

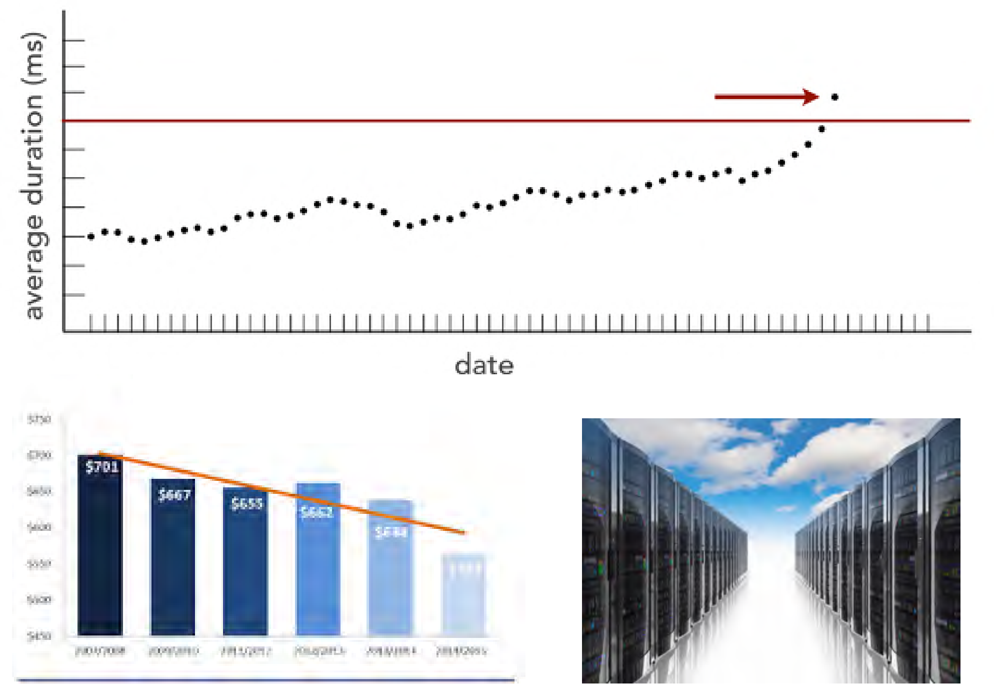


facilitate change

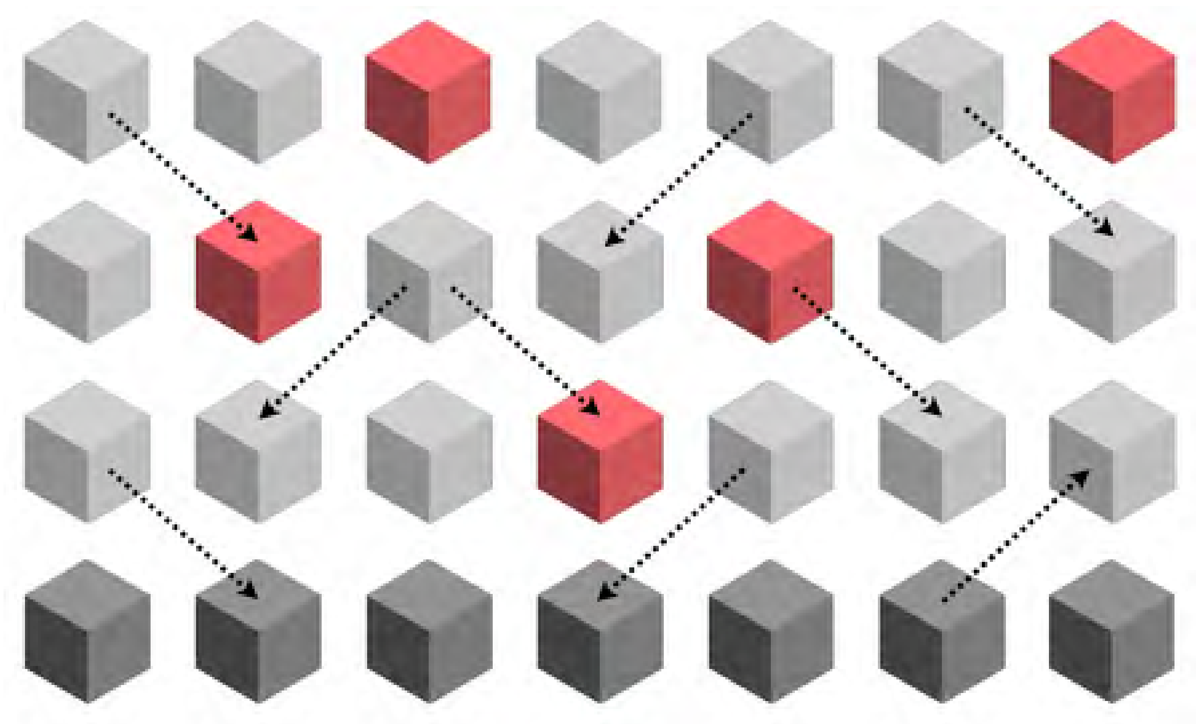
plan for change



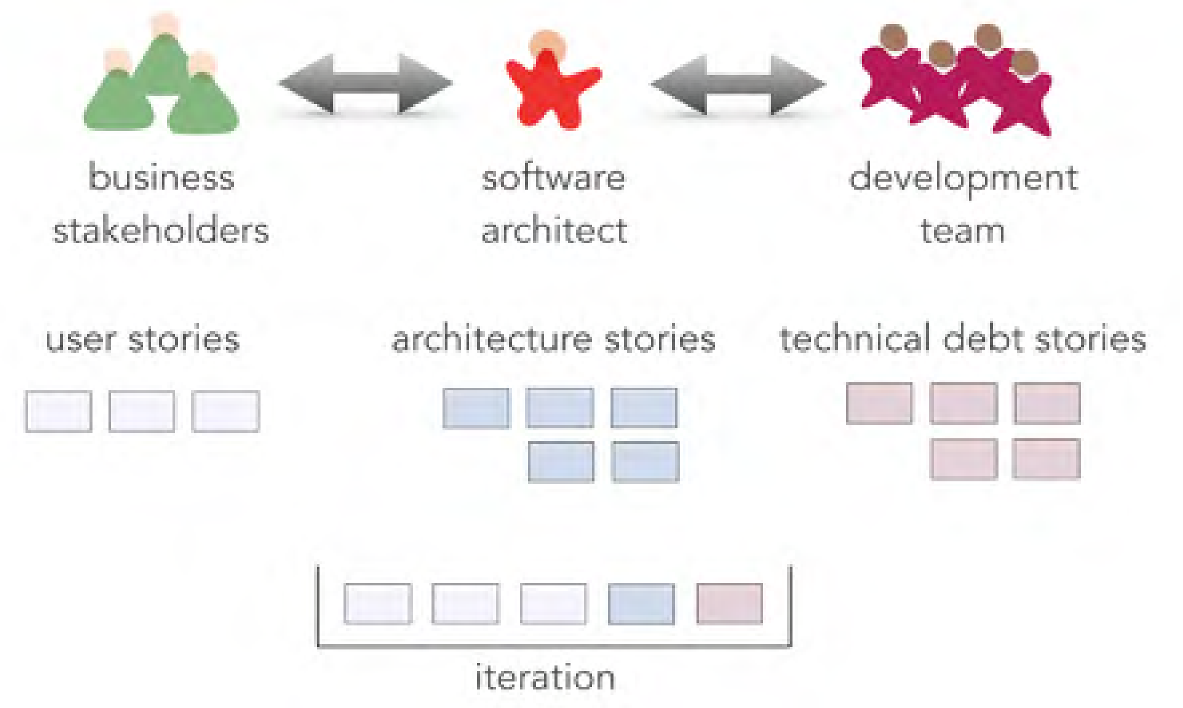
how do software architects handle all of this change?



detect change

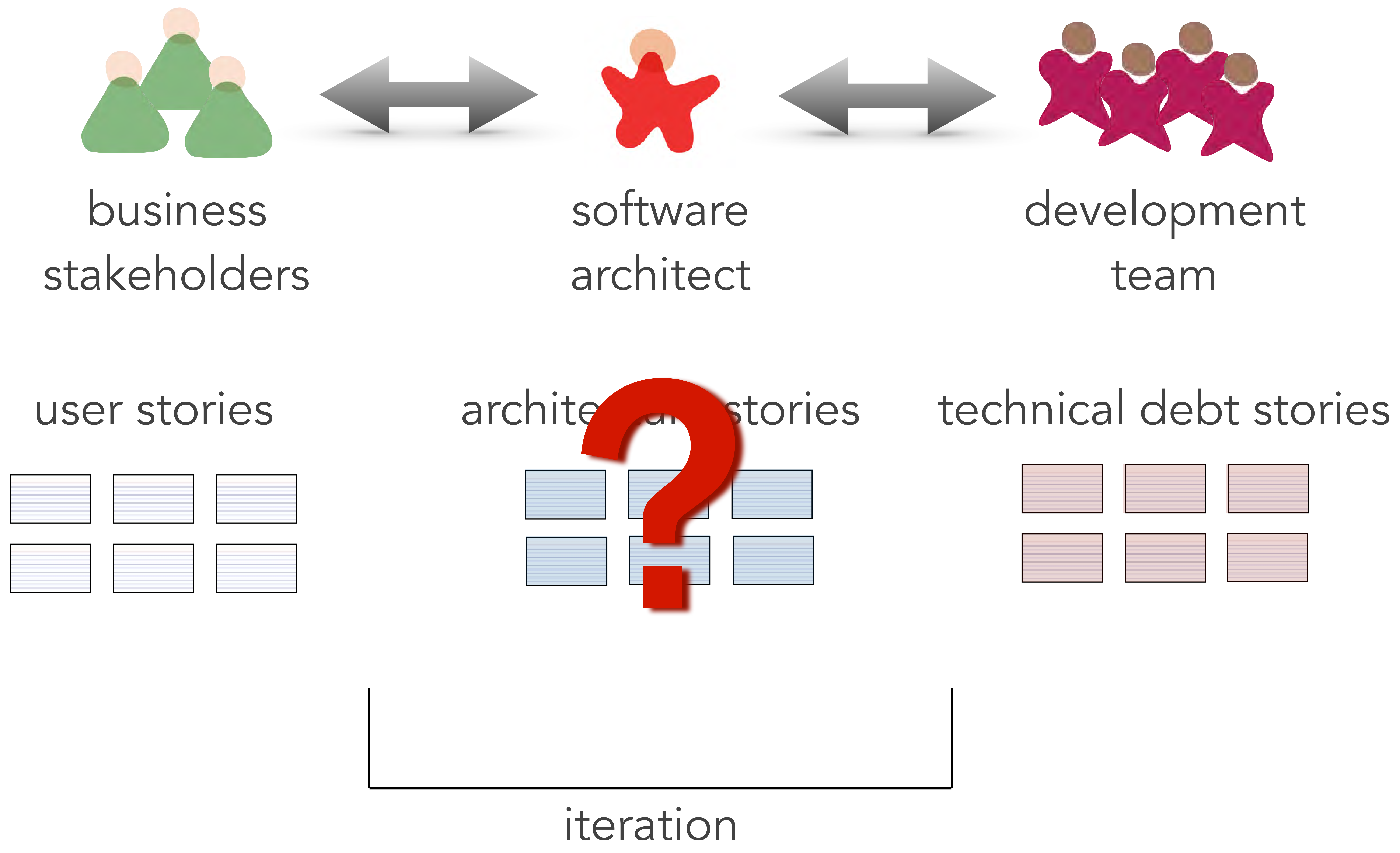


plan for change

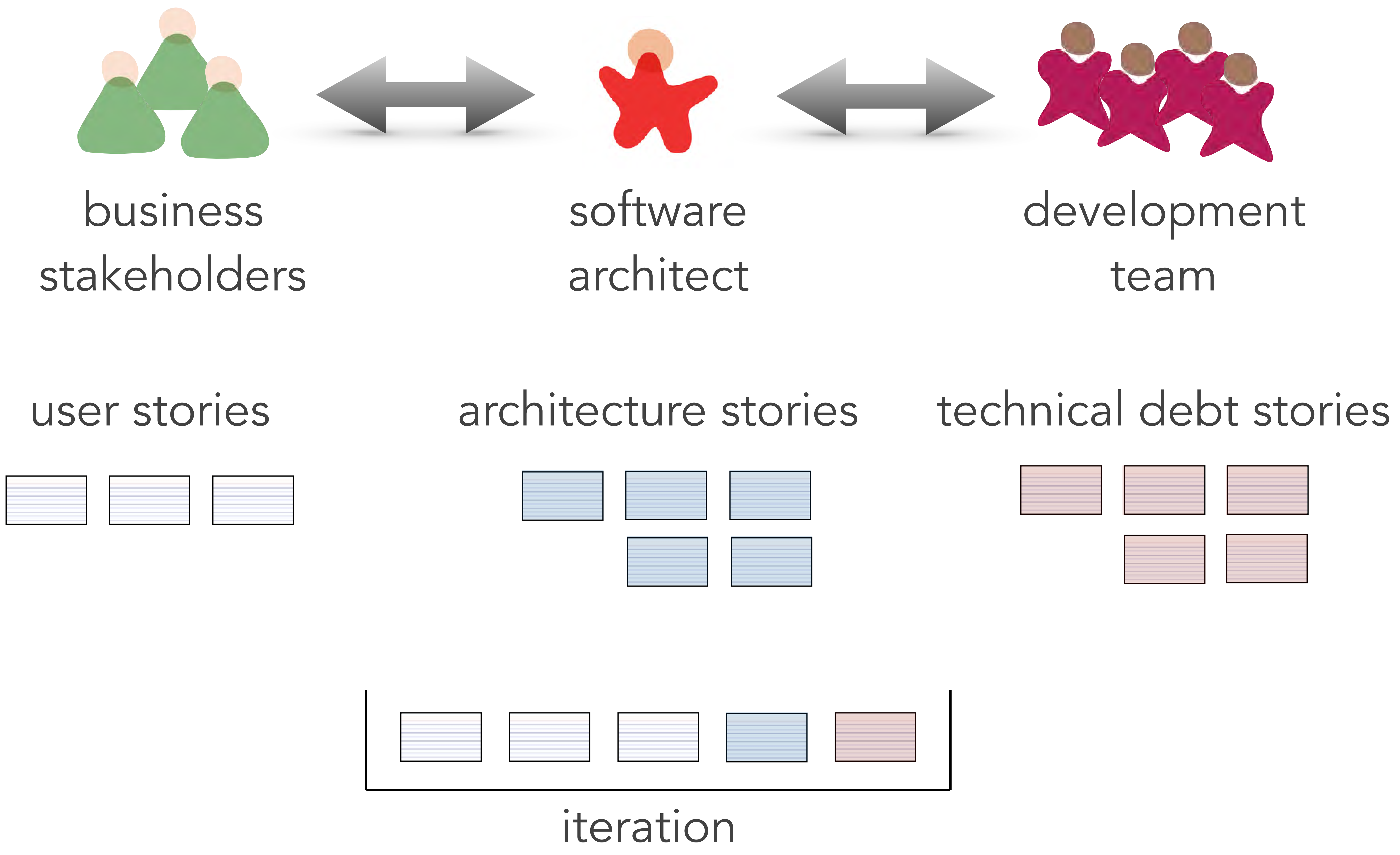


facilitate change

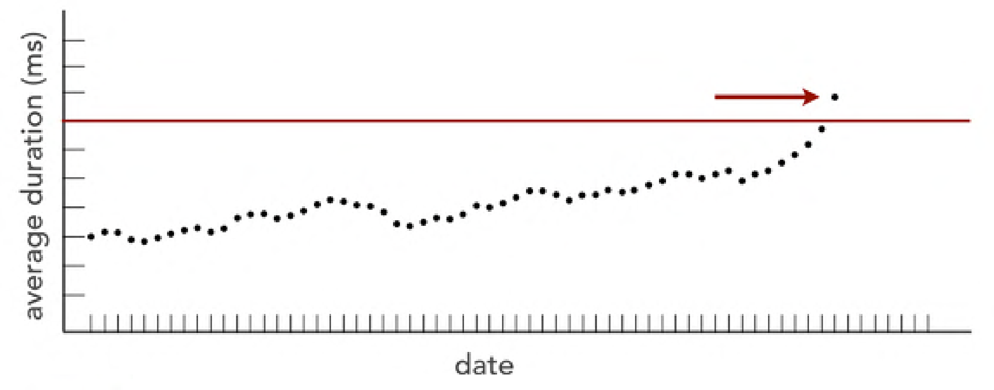
facilitate change



facilitate change

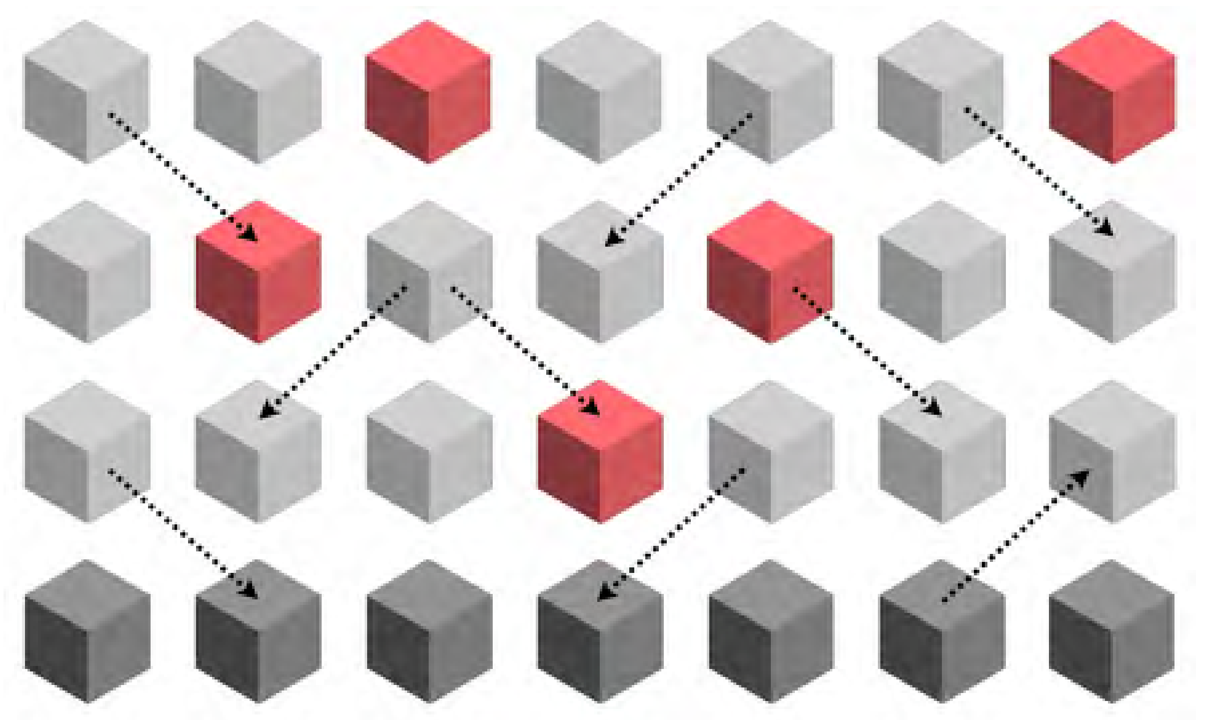


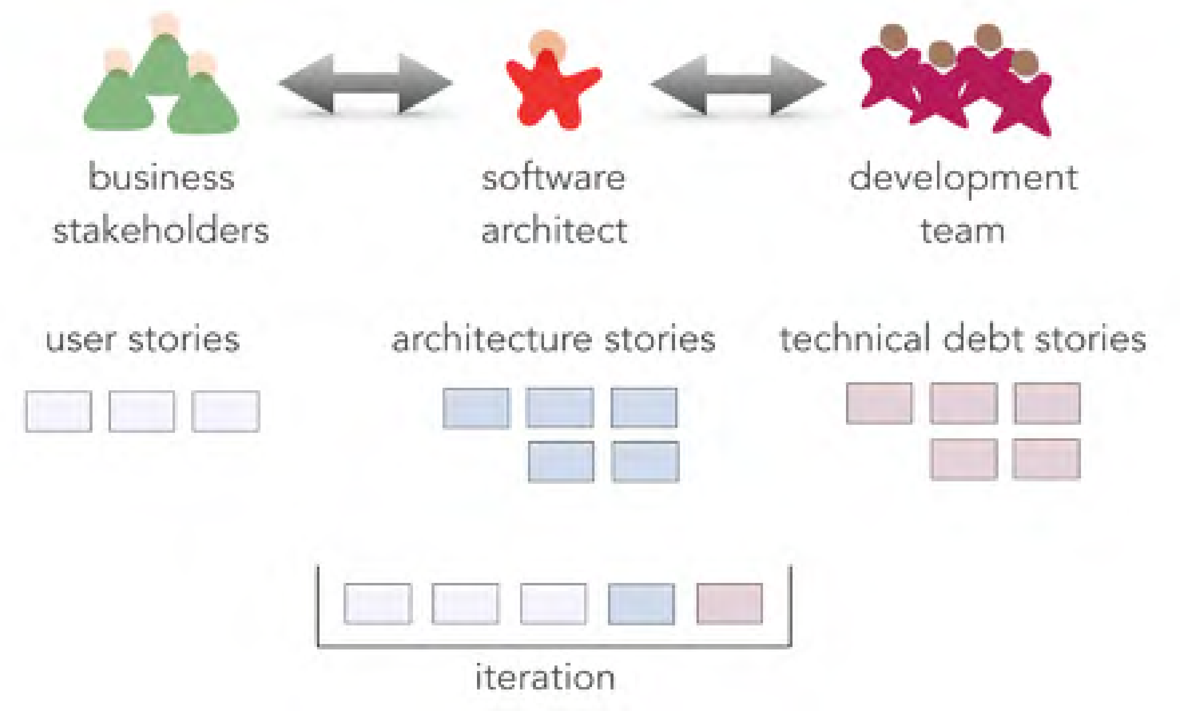
how do software architects handle all of this change?



$701
$667
$655
$662
$644

detect change


plan for change



facilitate change

O'Reilly software architecture keynotes are only 20 minutes long

Software Architecture Conference

Mark Richards

Independent Consultant

Hands-on Software Architect

Published Author|Conference Speaker

Architecture.Next: Invalidating Old Axioms

O'REILLY®