Domain Event Driven Architecture

Stefan Norberg,
Head of Architecture at Unibet.com

twitter: @stnor
email: stefan.norberg@unibet.com
blog: http://stnor.wordpress.com

@stnor

- 17 years as an IT professional
- Has worked with most aspects of IT
  - Operations & infrastructure
  - IT security
  - Systems development
  - Software architecture
  - Enterprise architecture
- Head of Architecture at Unibet

Agenda

- Basics and EDA intro
- The SOA problem
- How Domain EDA completes SOA
- Implementation notes from Unibet
- What to tell your manager

The three styles of interaction

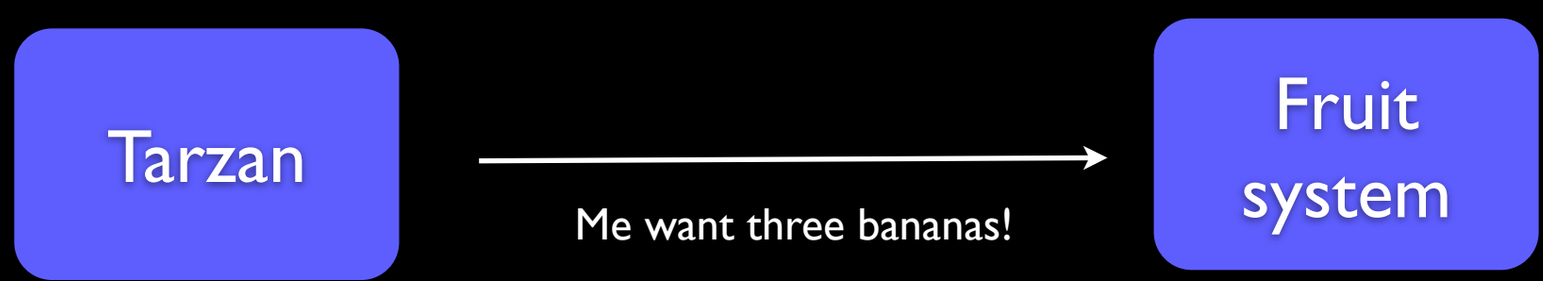
Type of interaction	*Initiator*	*Participants*
Time-driven	Time	The specified systems
Request-driven	Client	Client and Server
Event-driven	Event	Open-ended

Time driven

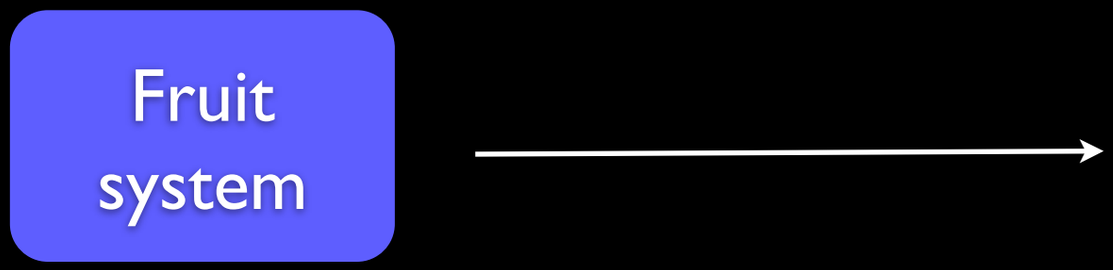
Fruit system

run inventory check every 60 mins

Request driven



Event driven



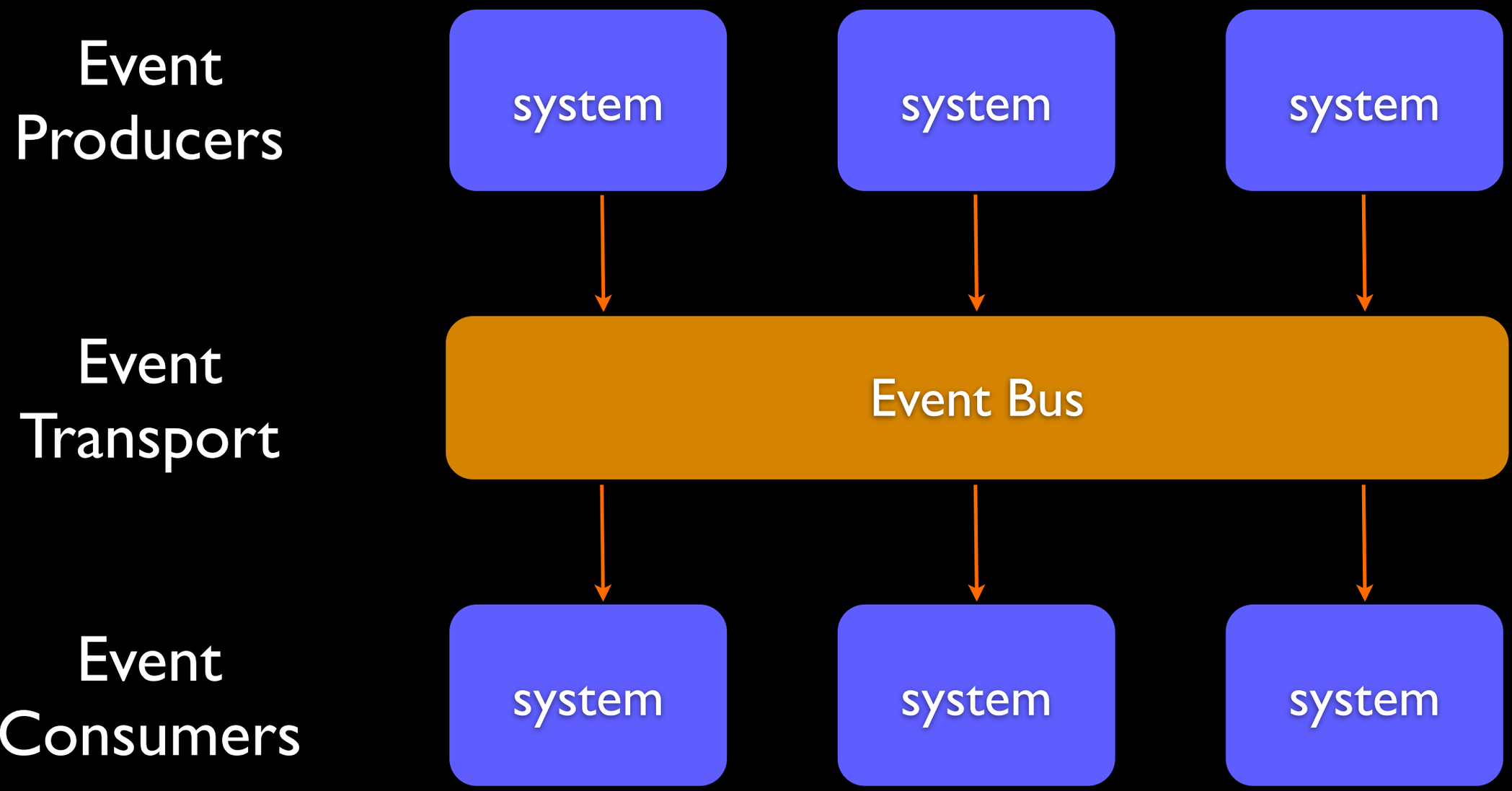
"Tarzan took three bananas"

"Fruit system is low on bananas"

5 Principles of EDA

- "Real-time" events as they happen at the producer
- Push notifications
- One-way "fire-and-forget"
- Immediate action at the consumers
- Informational ("someone logged in"), not commands ("audit this")

Typical EDA architecture

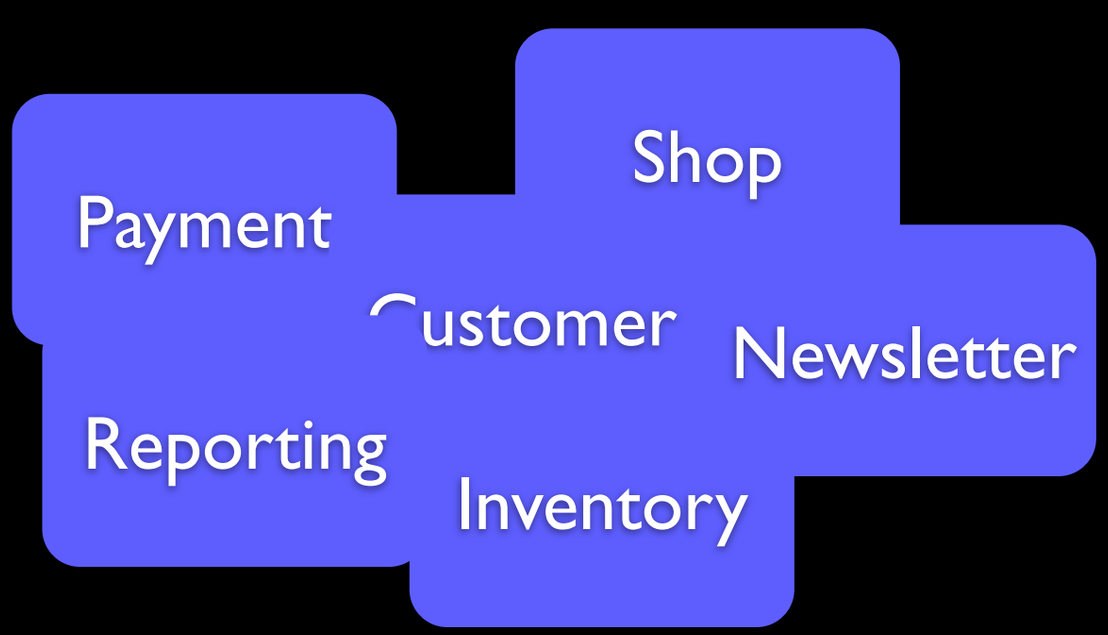


Main benefits of EDA

- Supports the business demands for better service (no batch, less waiting)
- No point-to-point integrations (fire & forget)
- Enables high performance, highly scalable systems

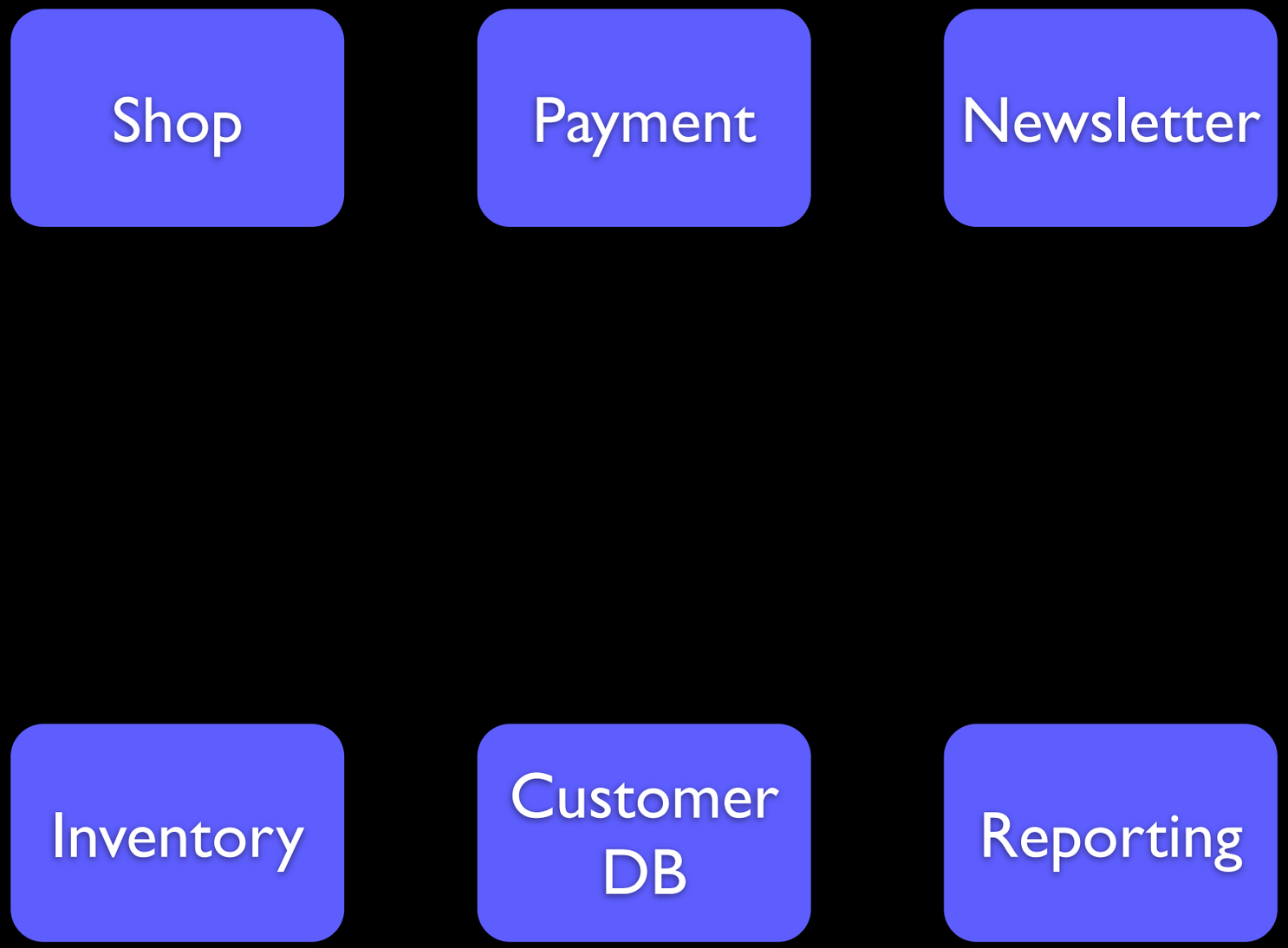
The birth of a system

The monolith



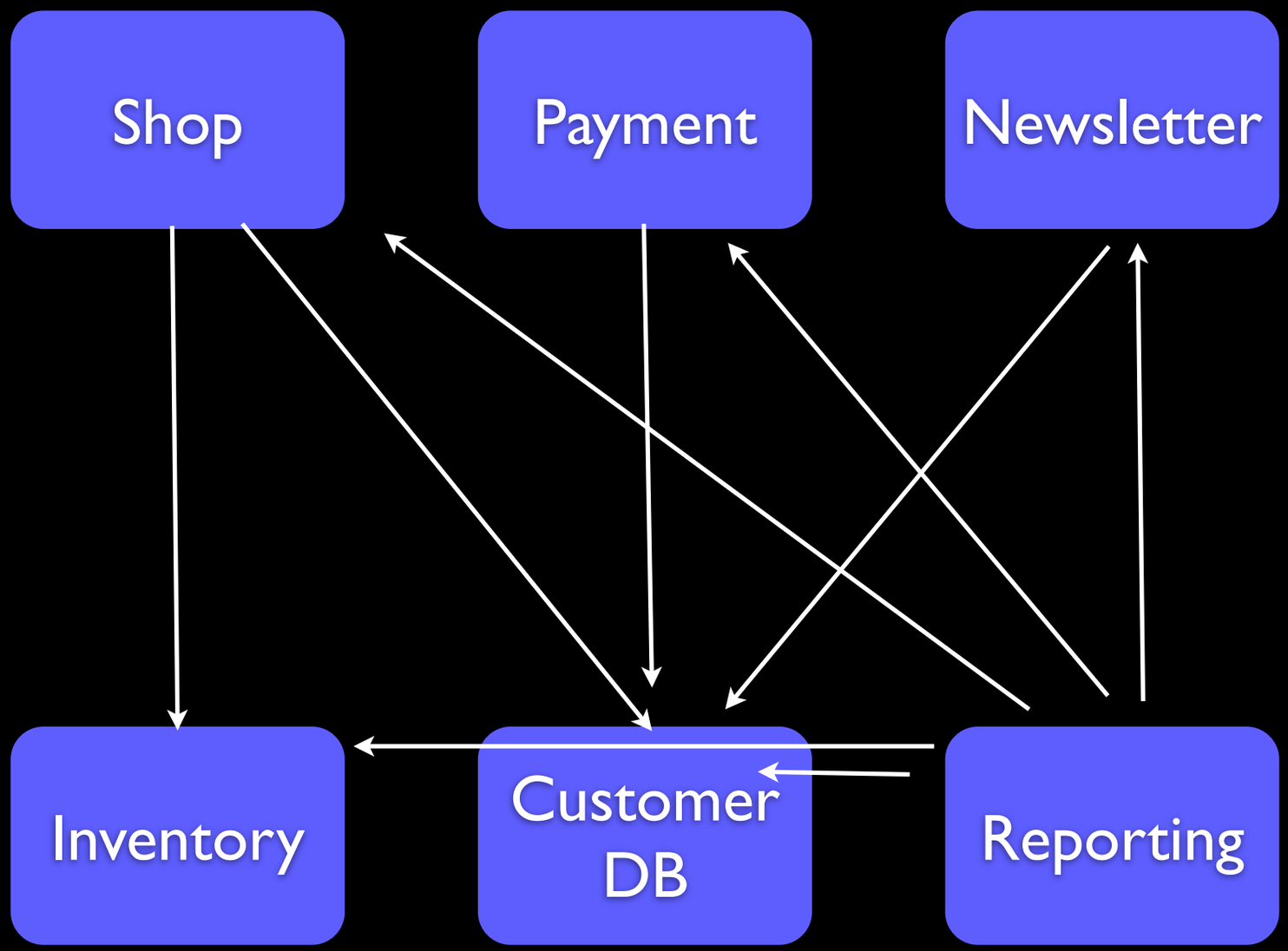
SOA architecture #1

Divide the problem domains into separate systems



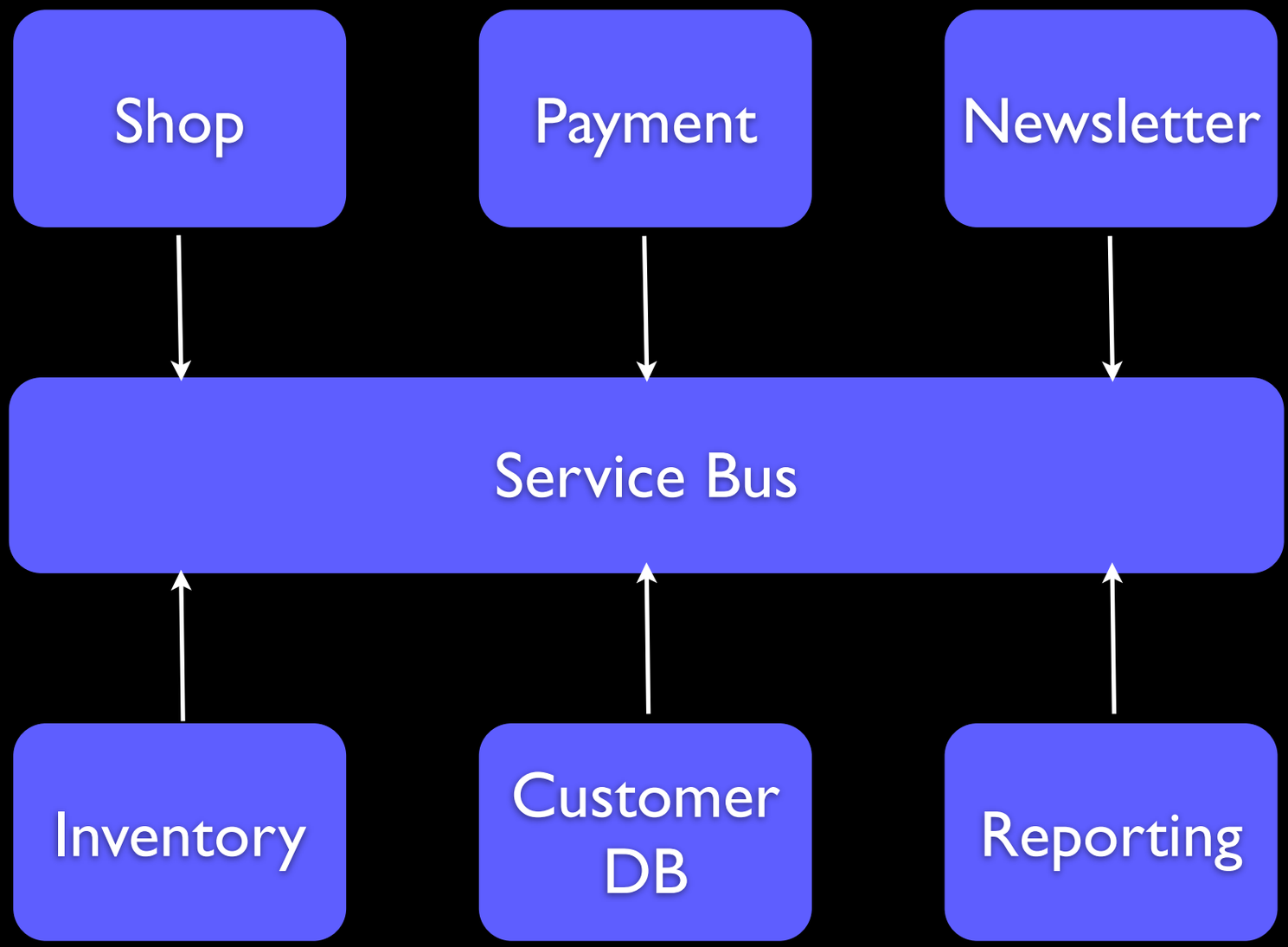
SOA architecture #1

A lot of point to point integration...

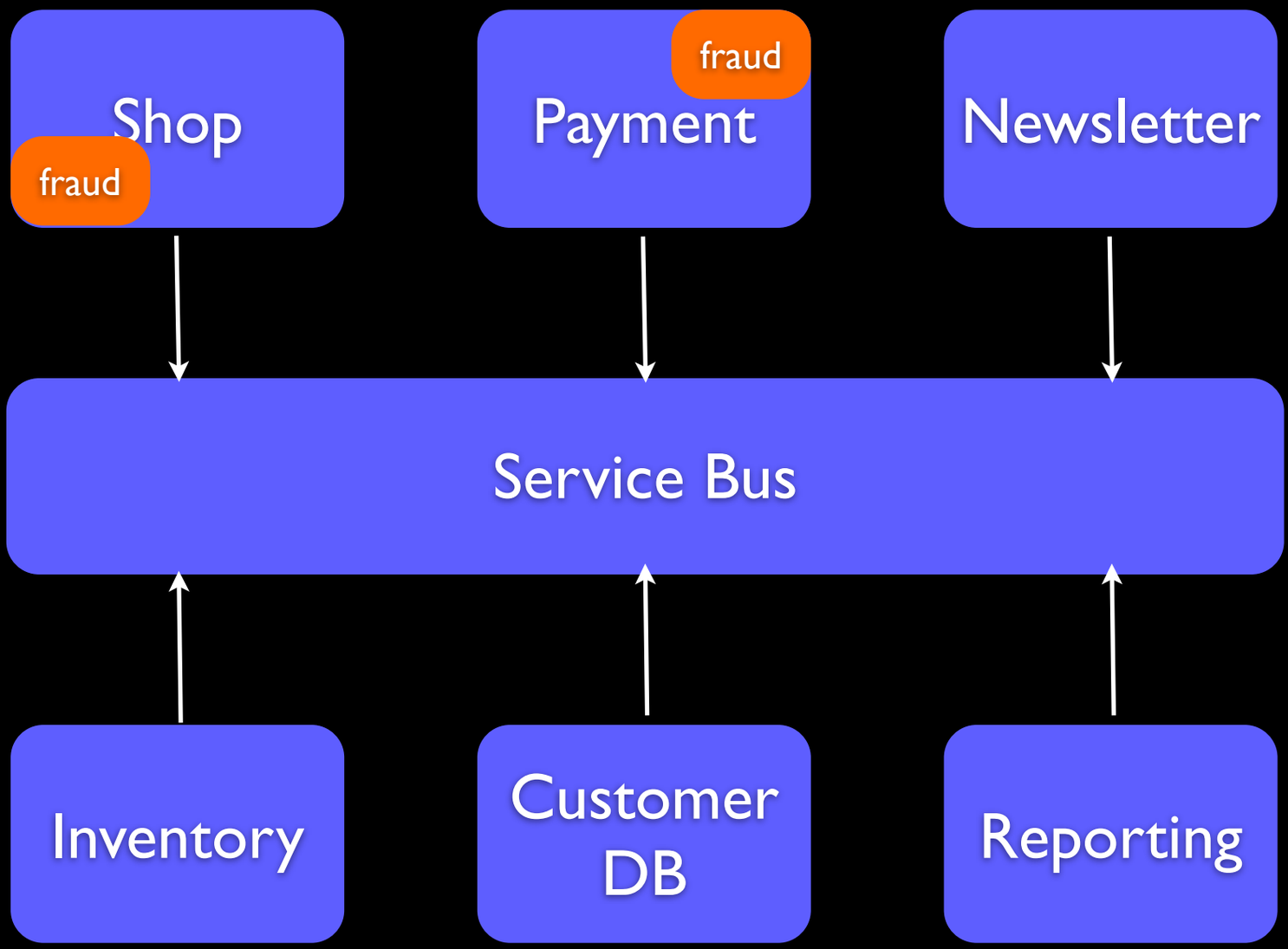


SOA architecture #2

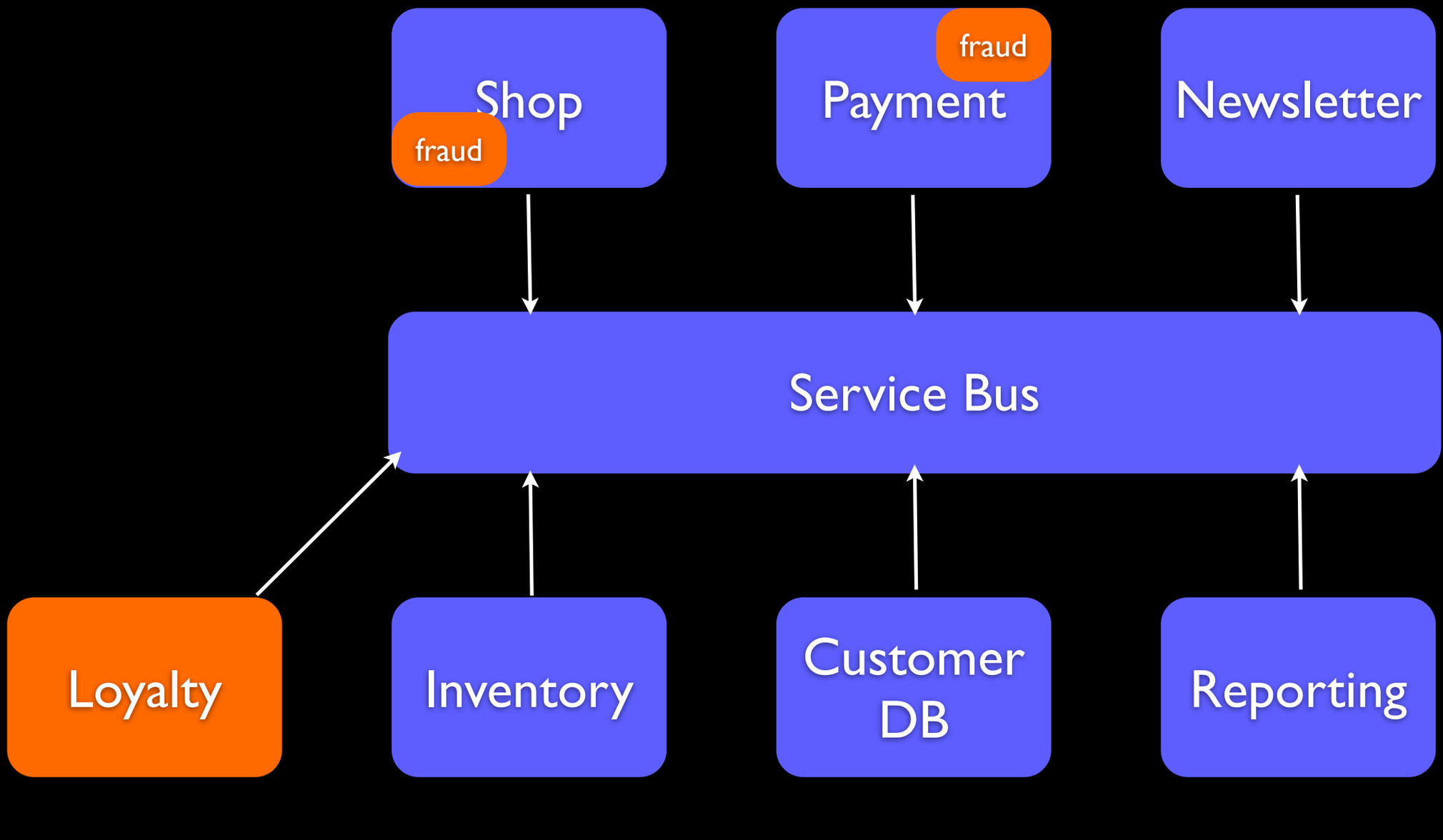
Monolith?



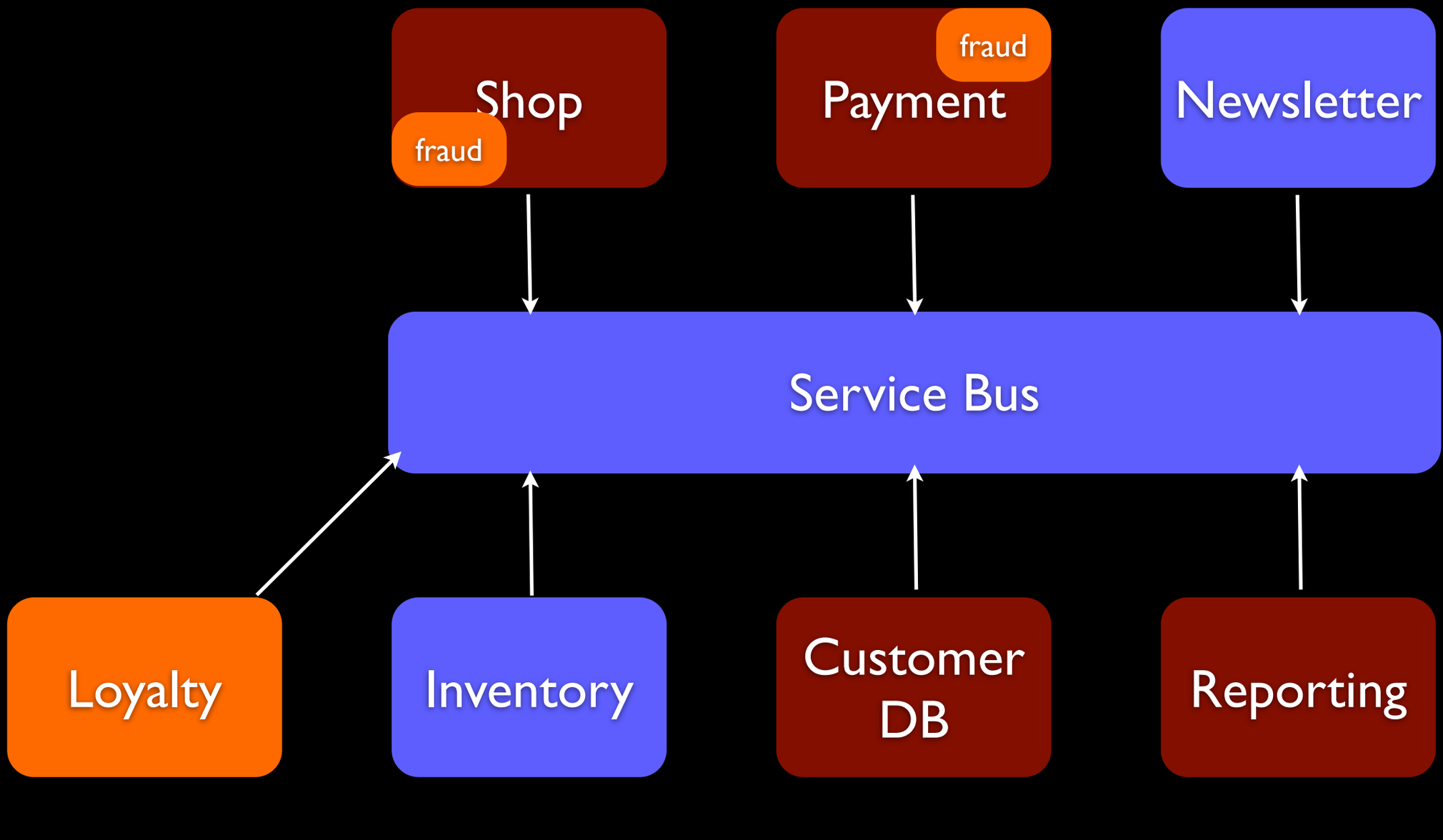
Let’s add fraud checks



Let's add a loyalty system



Integration ripple effect



Problem summary

- SOA is all about dividing domain logic into separate systems and exposing it as services
- Some domain logic will, by its very nature, be spread out over many systems
- The result is domain pollution and bloat in the SOA systems

“It's really become clear to me in the last couple of years that we need a new building block and that is the Domain Events”

-- Eric Evans, QCon 2009

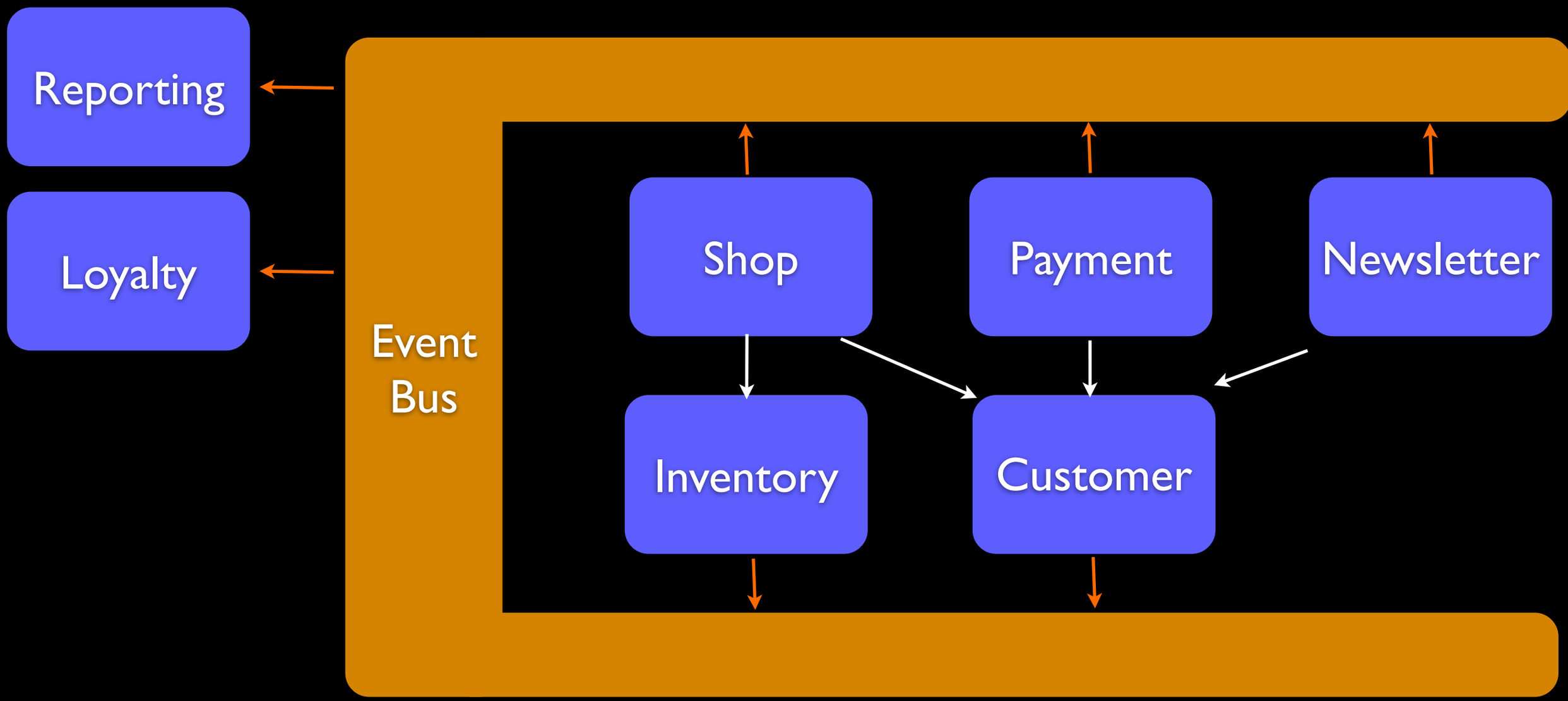
Domain Events and SOA

- A Domain Event is something that happened that the domain expert cares about
- A SOA system's primary focus should be on the domain and domain logic

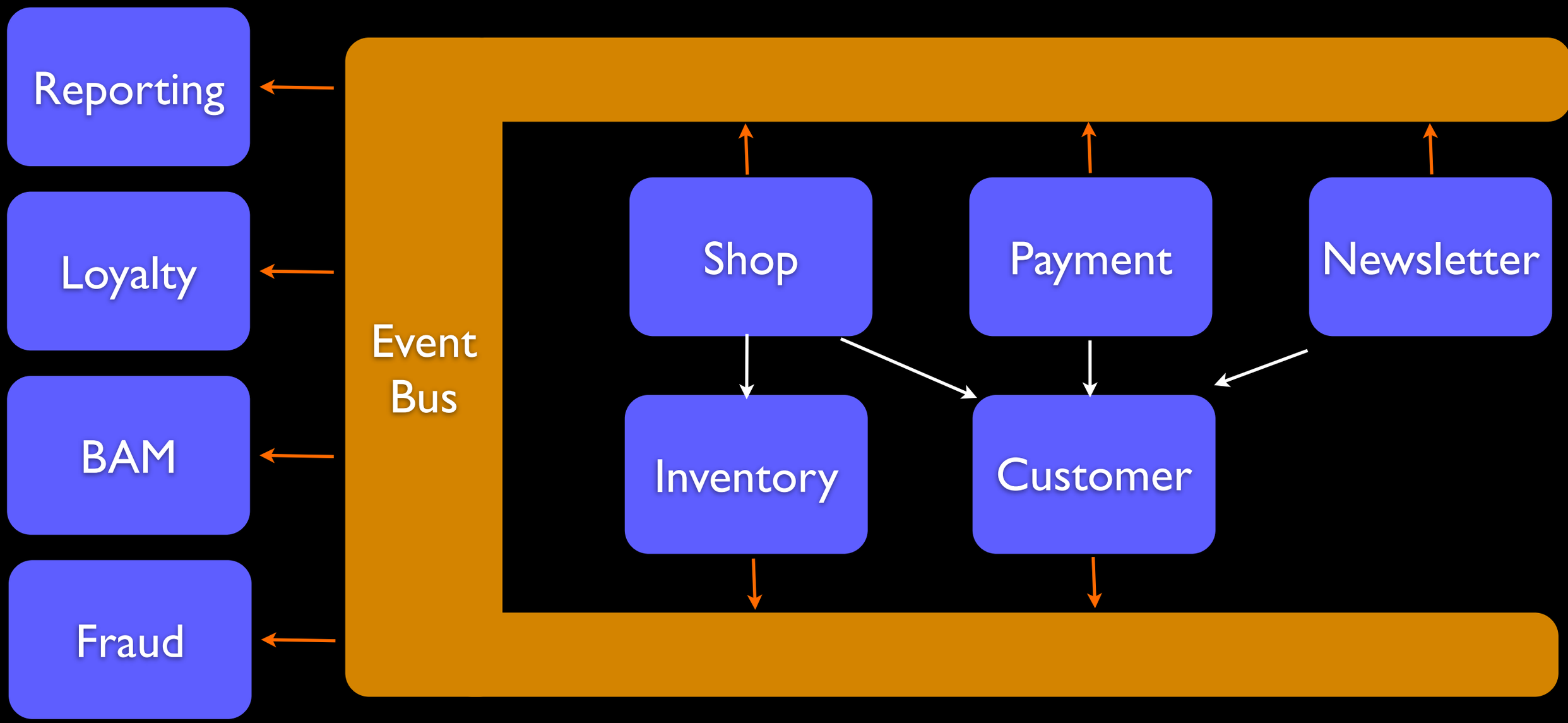
Domain EDA

- By exposing relevant Domain Events on a shared event bus we can isolate cross cutting functions to separate systems

Domain EDA + SOA



Domain EDA + SOA



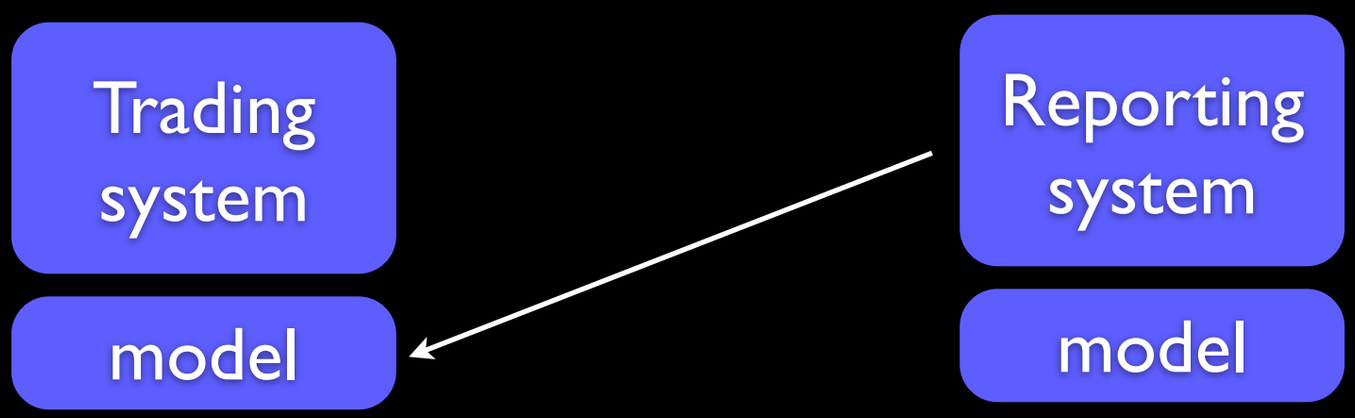
“You complete me”

Domain EDA

SOA

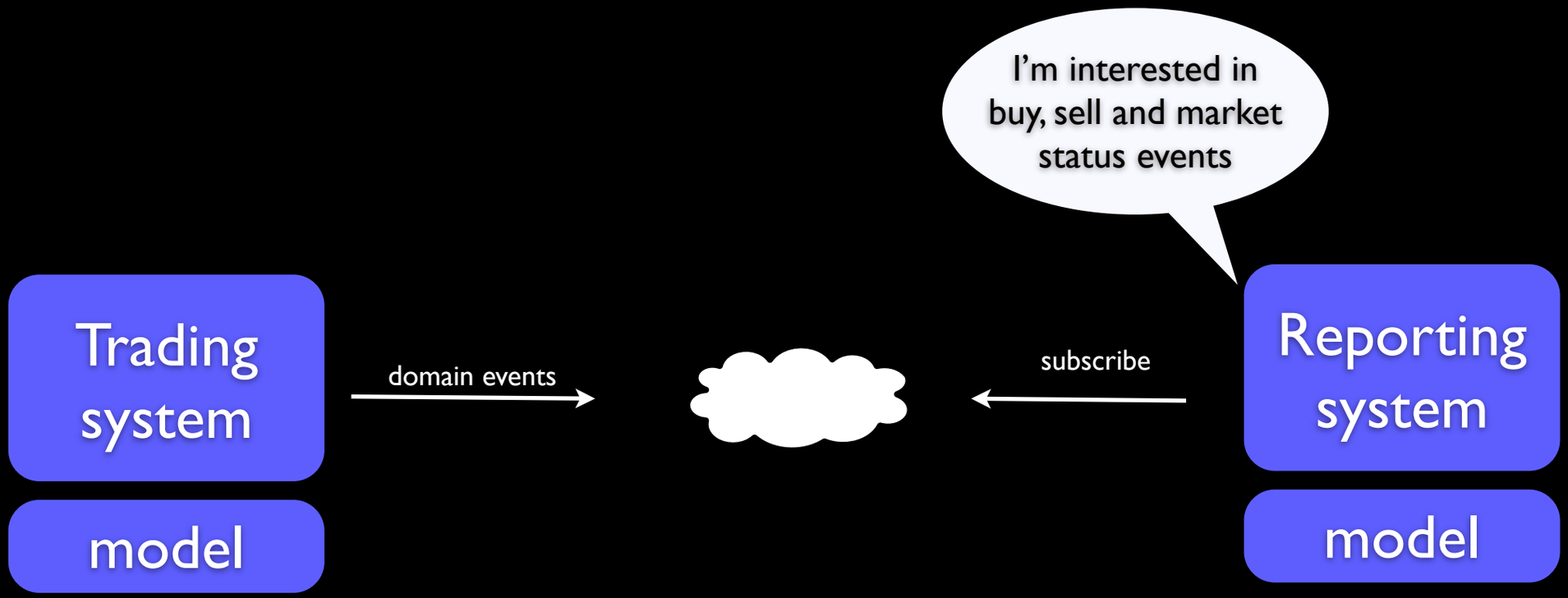
Example how Domain EDA decouples

Coupled integration

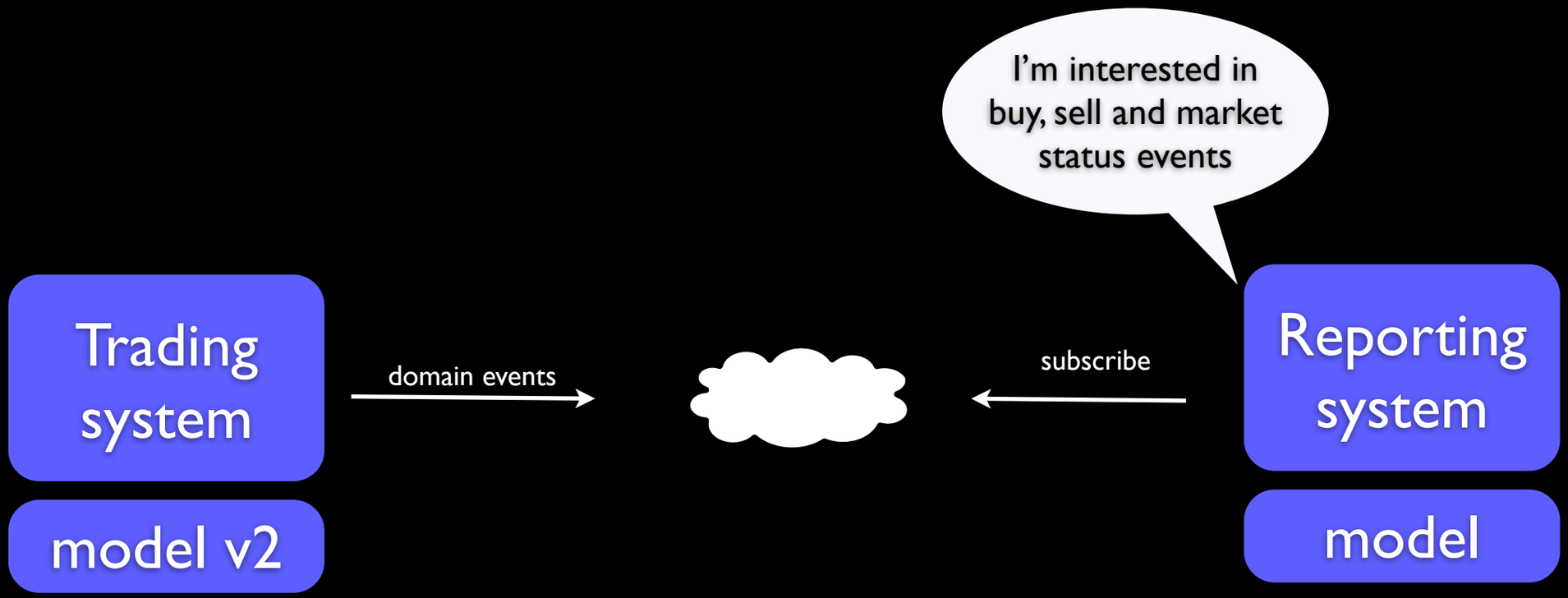


Coupled integration

Domain EDA integration



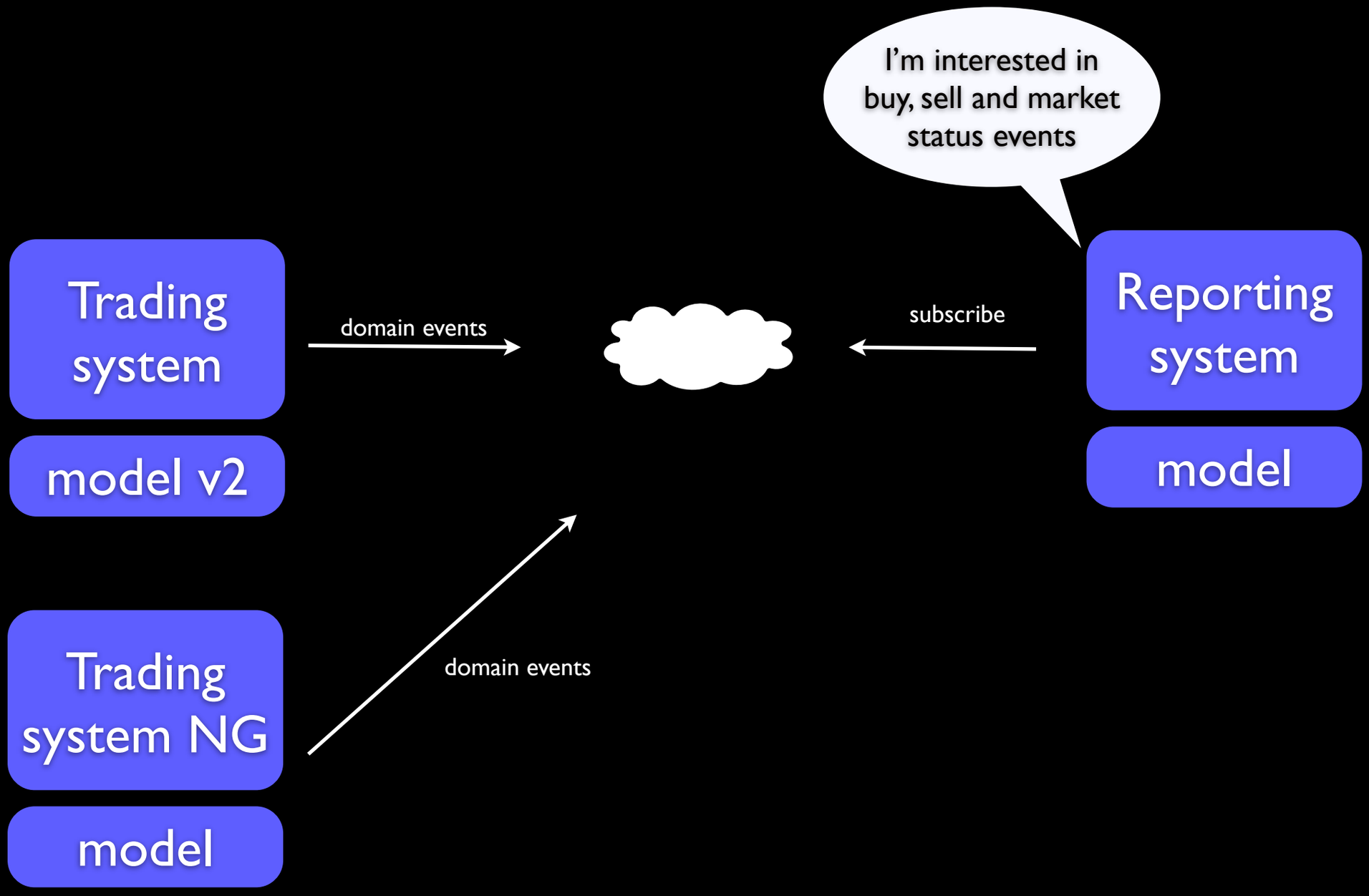
Domain EDA integration



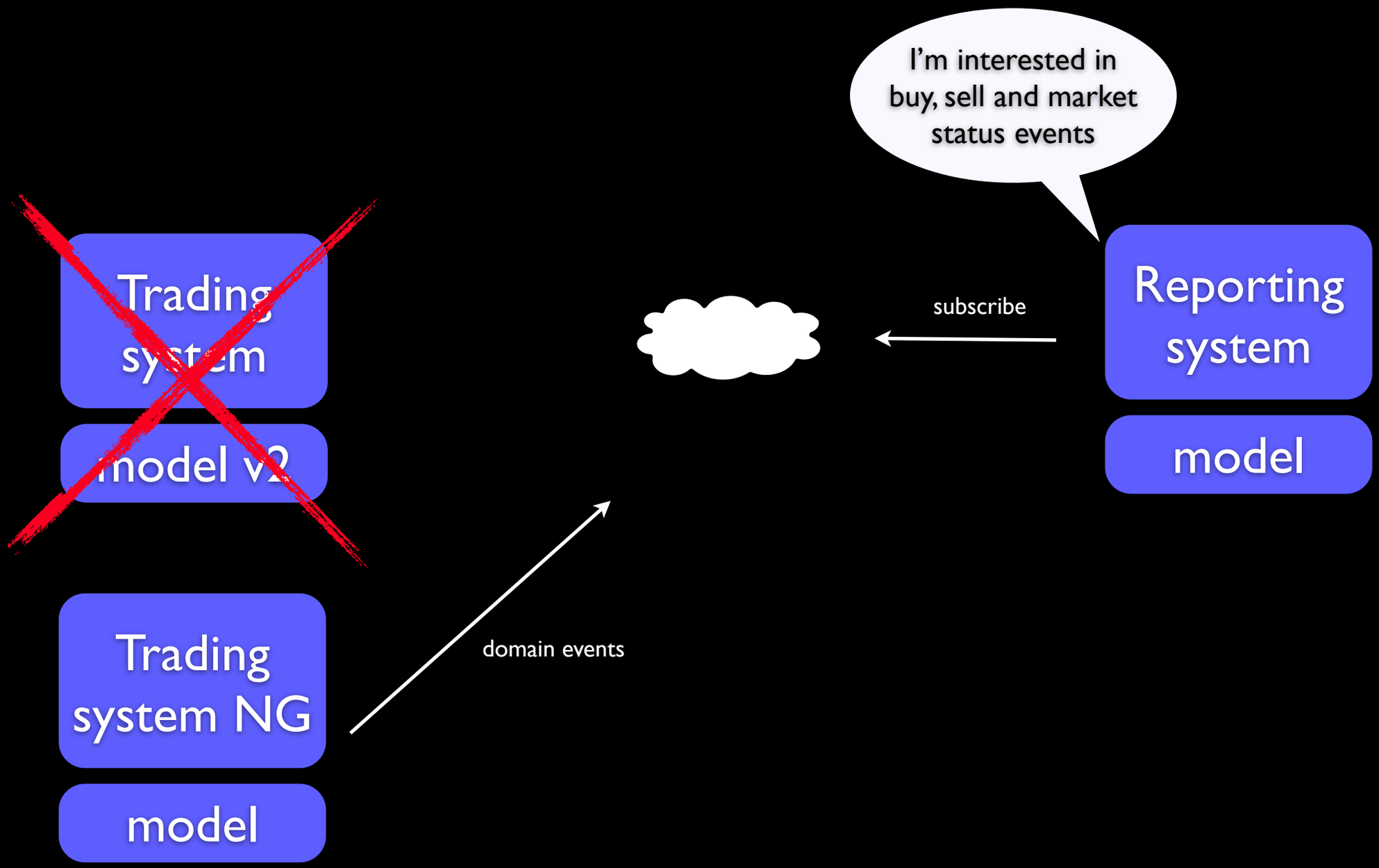
- “So what? I can do that with SOA...”

- “Yes, but can you do THIS?” (drumroll)

Domain EDA integration



Domain EDA integration



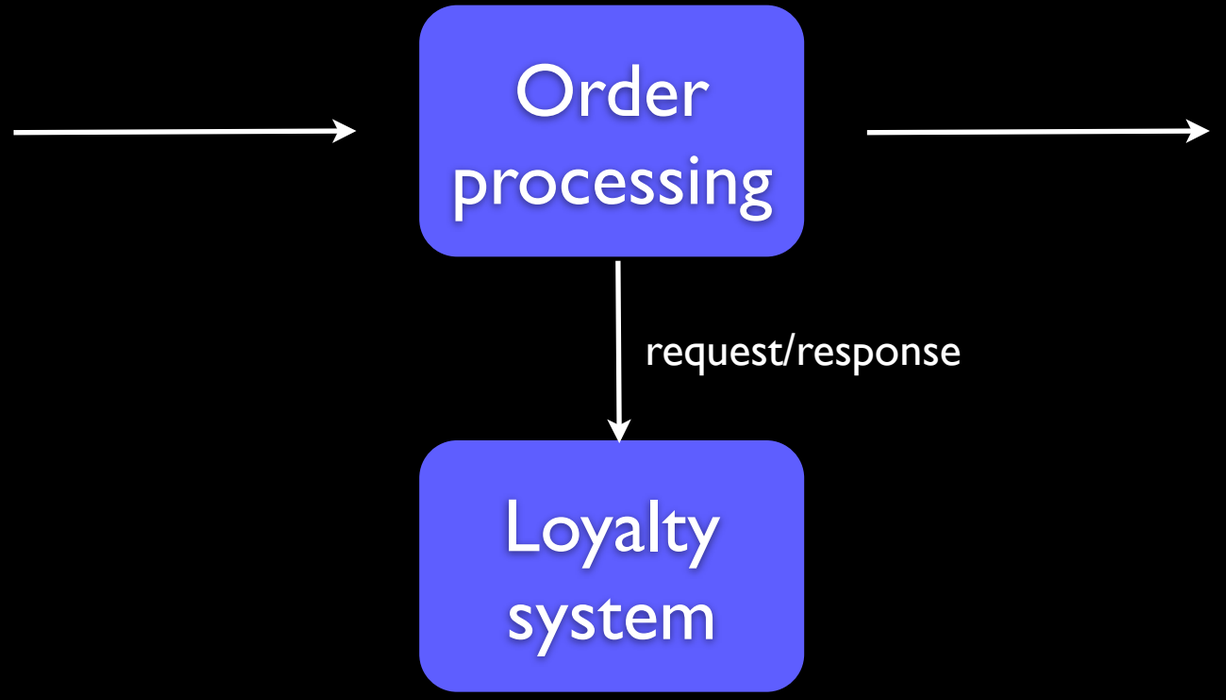
Example how EDA scales

Traditional SOA scalability issue

200 tx/s

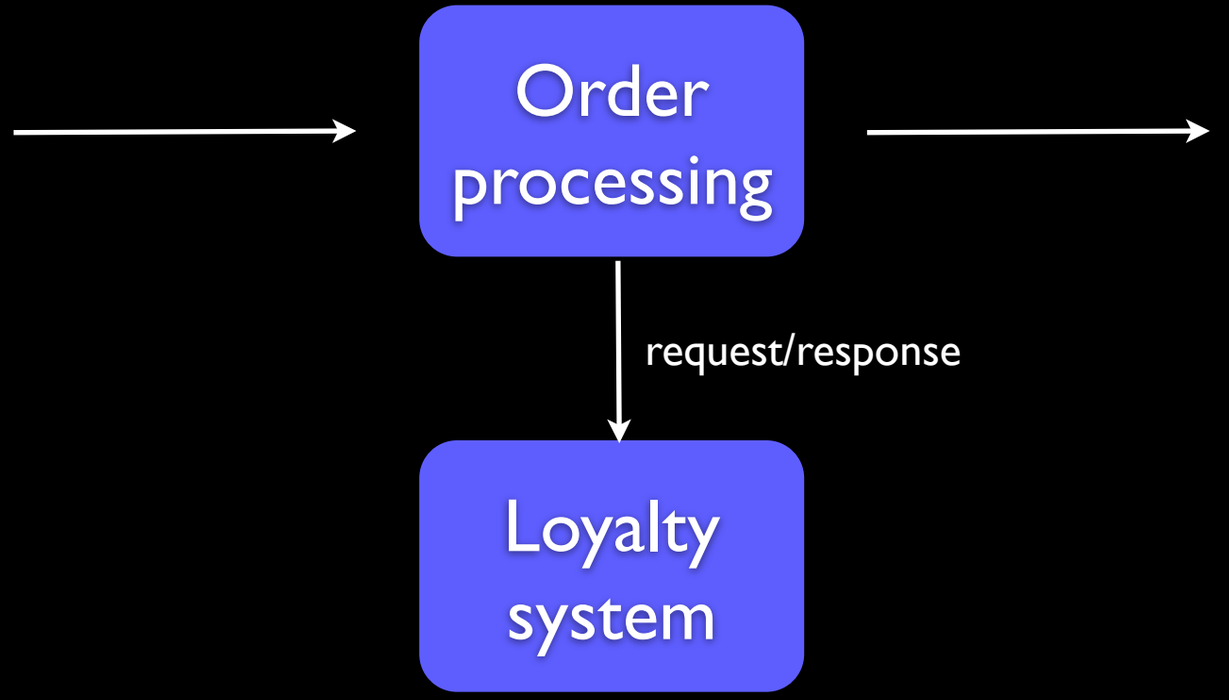
Traditional SOA scalability issue

200 tx/s



Traditional SOA scalability issue

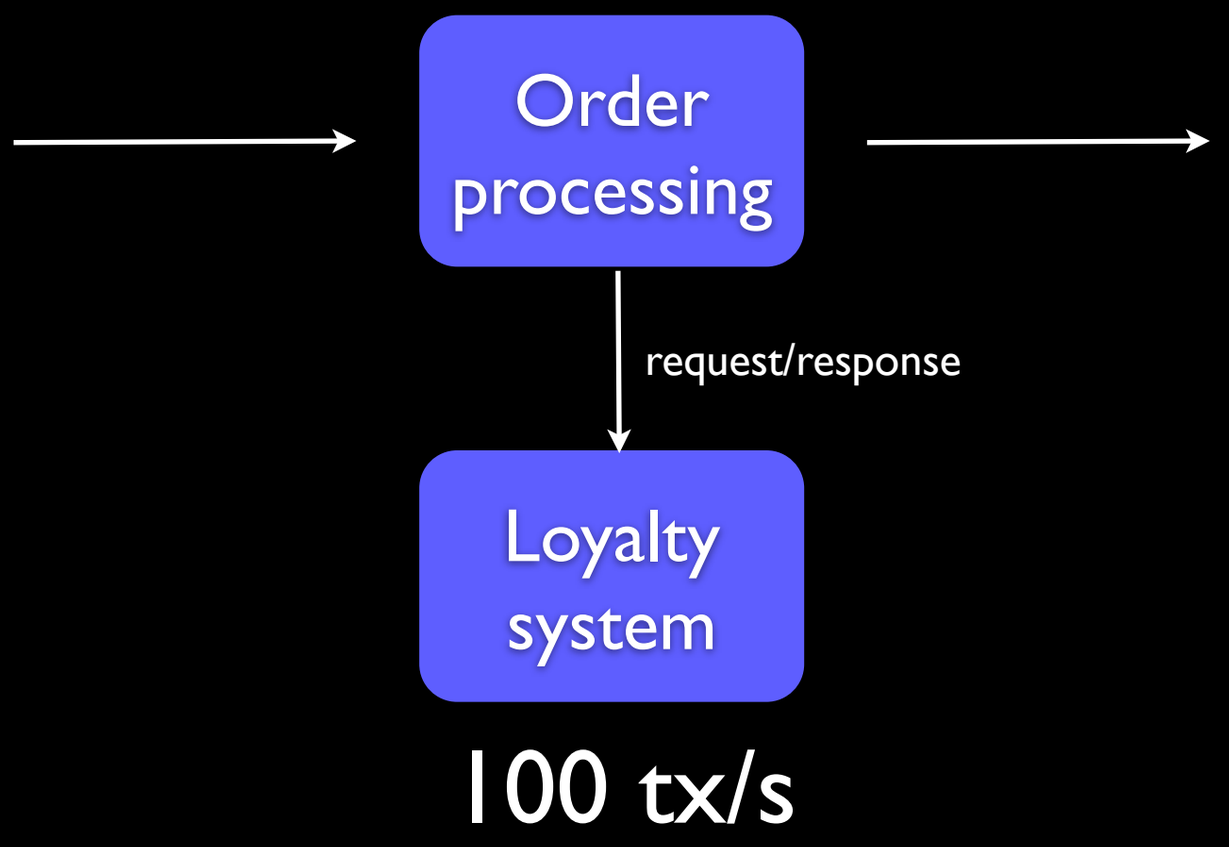
200 tx/s



100 tx/s

SOA: Weakest link dictates the
peak throughput
of the system

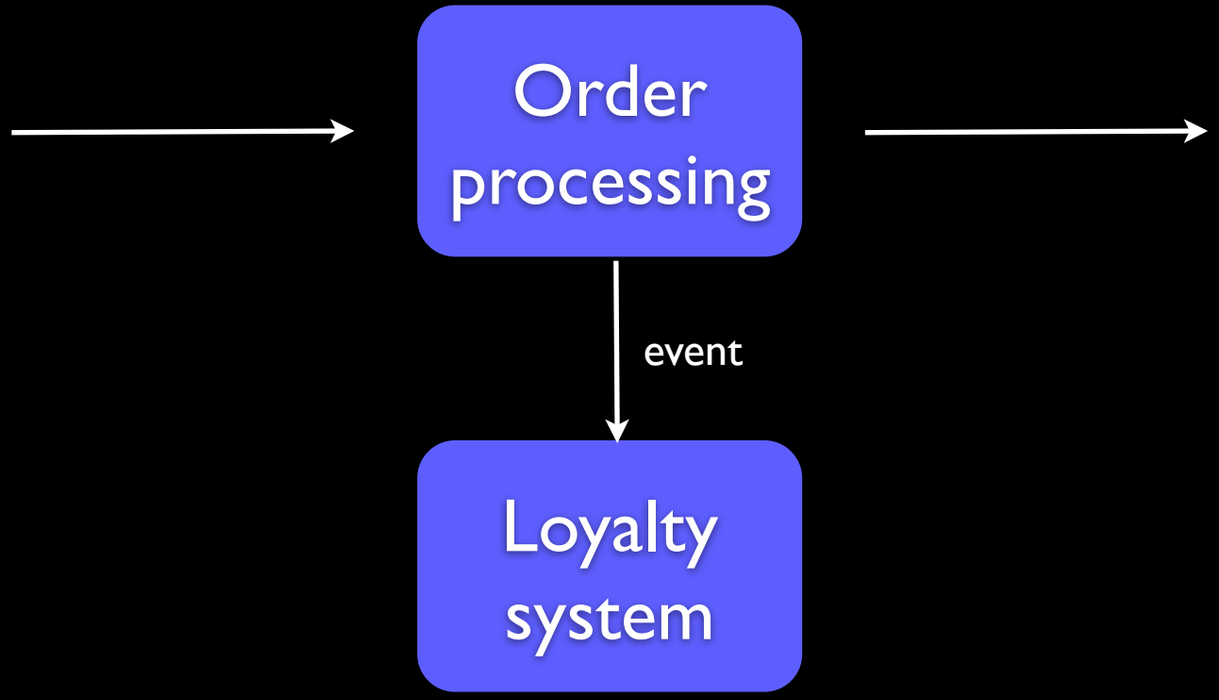
100 tx/s



100 tx/s

EDA: Weakest link dictates the
sustained throughput
of the system

200 tx/s



100 tx/s

Implementation notes from Unibet

hans.hall@unibet.com

MQ considerations

- Ordering
- Durability
- Performance
- Once and only once delivery

Our requirements

- 1000 messages / second sustained
- Guaranteed delivery
- Easy to use client

Brokers tested

- Lots of brokers tested
  - ActiveMQ, OpenMQ, HornetQ, Websphere, Fiorano, RabbitMQ, QPID
- Second round
  - ActiveMQ, OpenMQ and RabbitMQ
- Finally went for ActiveMQ - fitted our requirements/use case the best

Testing brokers

- Killing brokers
- Flow control
- Slow consumers
- Durable / non-durable
- Small configuration change can have huge impact on throughput

Payload considerations

The smorgasbord

Name	*Standardized*	*Schema support*	*Binary*	*Easy to use*	*X-platform*	*Std API*
ASN.1	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**No**
CSV	**Partial**	**No**	**No**	**No?**	**Yes**	**No**
XML	**Yes**	**Yes**	**No**	**Yes**	**Yes**	DOM, SAX, XQUERY, XPATH
JSON	**Yes**	**No**	**No**	**Yes**	**Yes**	**No**
Java serialization	**No**	**Partial**	**Yes**	**Yes**	**No**	**No**
Hessian	**No**	**Partial**	**Yes**	**Yes**	**Yes**	**No**
Protocol Buffers	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**No**
BSON	**No**	**No**	**Yes**	**Yes**	**Yes**	**No**

Domain events need schema support

Name	*Standardized*	*Schema support*	*Binary*	*Easy to use*	*X-platform*	*Std API*
ASN.1	Yes	**Yes**	Yes	No	Yes	No
CSV	Partial	**No**	No	No?	Yes	No
XML	Yes	**Yes**	No	Yes	Yes	DOM, SAX, XQUERY, XPATH
JSON	Yes	**No**	No	Yes	Yes	No
Java serialization	No	**Partial**	Yes	Yes	No	No
Hessian	No	**Partial**	Yes	Yes	Yes	No
Protocol Buffers	No	**Yes**	Yes	Yes	Yes	No
BSON	No	**No**	Yes	Yes	Yes	No

+ Efficient, x-platform and easy to use

Name	*Standardized*	*Schema support*	*Binary*	*Easy to use*	*X-platform*	*Std API*
ASN.1	Yes	**Yes**	**No**	**No**	**Yes**	No
XML	Yes	**Yes**	**No**	**Yes**	**Yes**	DOM, SAX, XQUERY, XPATH
Protocol Buffers	No	**Yes**	**Yes**	**Yes**	**Yes**	No

Protocol Buffers

- Flexible, efficient, automated mechanism for serializing (binary)
- Schema support (.proto files)
- Language neutral
- Invented and used by Google Open Sourced in July 2008

Example

newuserevent.proto:

```
option java_package = "com.example.foo";
message NewUserEvent {
  required string name = 1;
  required int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    required string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phone = 4;
}
```

Generating Java Source

```
$ protoc --proto_path=IMPORT_PATH \
         --java_out=DST_DIR \
         path/to/file.proto
```

IMPORT_PATH specifies a directory in which to look for .proto files when resolving import directives.

If the **DST_DIR** ends in .zip or .jar, the compiler will write the output to a single ZIP-format archive file with the given name.

Builders vs messages

- Messages classes are immutable
- Use builders to construct

```
NewUserEvent stefan =
  NewUserEvent.newBuilder()
    .setId(1337)
    .setName("Stefan Norberg")
    .setEmail(“stefan@norberg.org")
    .addPhone(
      NewUserEvent.PhoneNumber.newBuilder()
        .setNumber("+46 70 99 66 547")
        .setType(NewUserEvent.PhoneType.WORK))
    .build();
```

Evolving messages

- You cannot remove required fields
- New fields that you add should be optional or repeated
- New fields should have sensible defaults
- Prefix deprecated non-required fields with OBSOLETE_ (convention)

Sample EDA producer

```
@Autowired
DomainEventPublisher publisher;

NewUserEvent stefan =
  NewUserEvent.newBuilder()
    .setId(1337)
    .setName("Stefan Norberg")
    .setEmail(“stefan@norberg.org")
    .addPhone(
      NewUserEvent.PhoneNumber.newBuilder()
        .setNumber("+46 70 99 66 547")
        .setType(NewUserEvent.PhoneType.WORK))
    .build();

publisher.publish(stefan);
```

Sample EDA consumer

```
public class YourDomainListenerPOJO {
    public void receive(LoginEvent loginEvent) {
      // do something
    }

    @Durable(clientId="com.example.foo.bar")
            public void receive(NewUserEvent newUserEvent) {
              // do something
            }
        }
```

```
<import resource="classpath:spring-domain-events.xml" />

<bean id="domainEventListener"
      class="com.foo.YourDomainListenerPOJO"/>

<domain-events:subscribe id="eventSubscriber"
                         subscriber="domainEventListener"/>
```

What to tell your manager

What to tell your manager

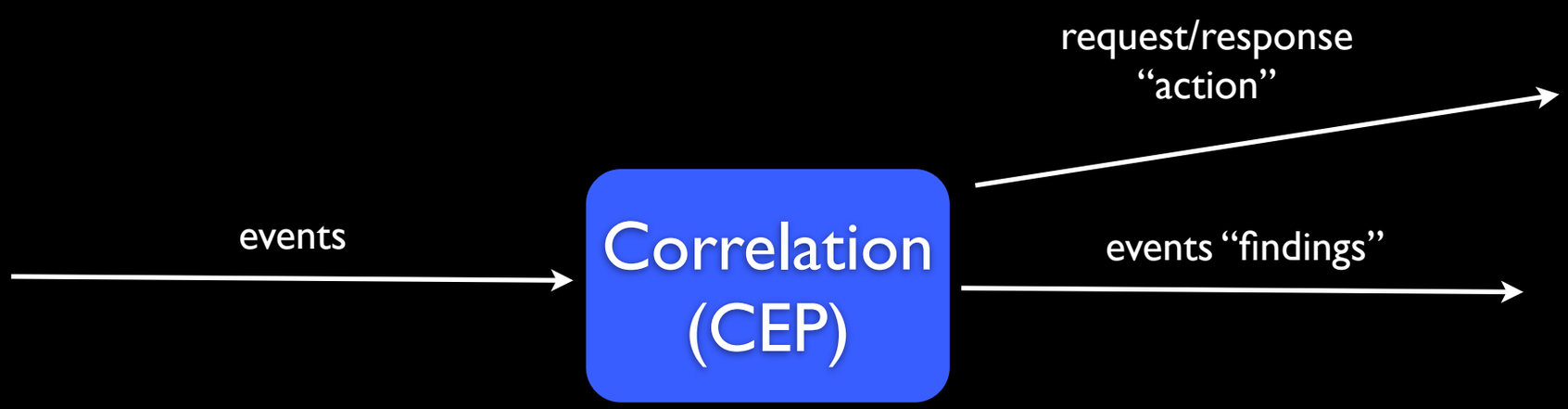
- SOA+EDA will reduce time-to-market for new functionality
- SOA+EDA will enable a layer of high-value services that have a visible impact on the bottom line of the business

Game changing
value-add

Complex Event Processing (CEP)

- 100% Event-driven
- Receives domain events
- Performs ***event correlation*** using a QL
- Fires domain events ("findings")

CEP Examples



if there are no Mastercard deposits from France in 5 minutes during business hours,
send NoDespositsEvent

if there are >30 failed logins using >5 accounts from the same ip within 2 minutes,
block the ip for 24 hours

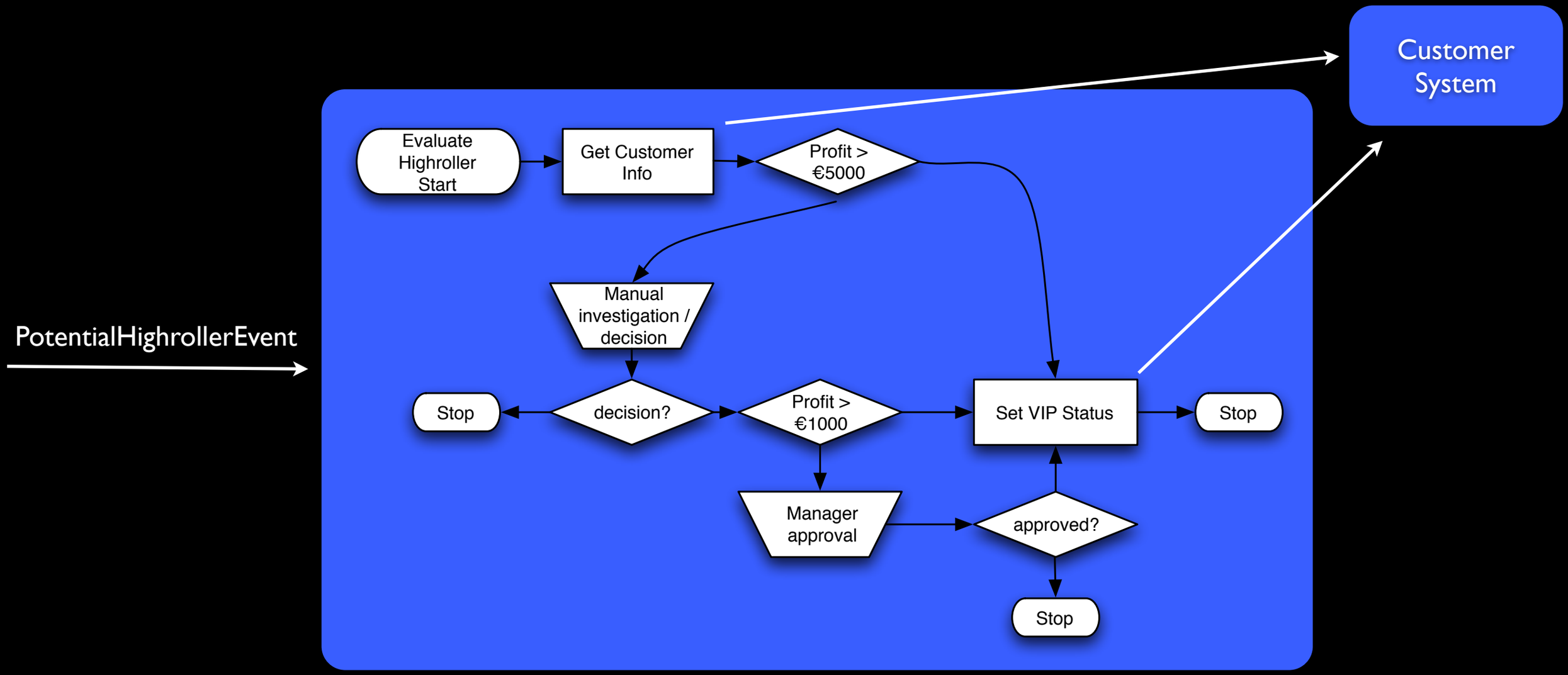
if customer with loyalty >= gold and puts goods in shopping cart for more than €200,
send VIPShoppingEvent

if customer loses €2000 in the casino within 30 minutes and customer != highroller
send PotentialHighrollerEvent AND grant €200 casino premium

Business Process Management (BPM)

- Business control over definition and automation of business processes
- SOA services "orchestration"
- Processes can/should be started by domain events

BPM example

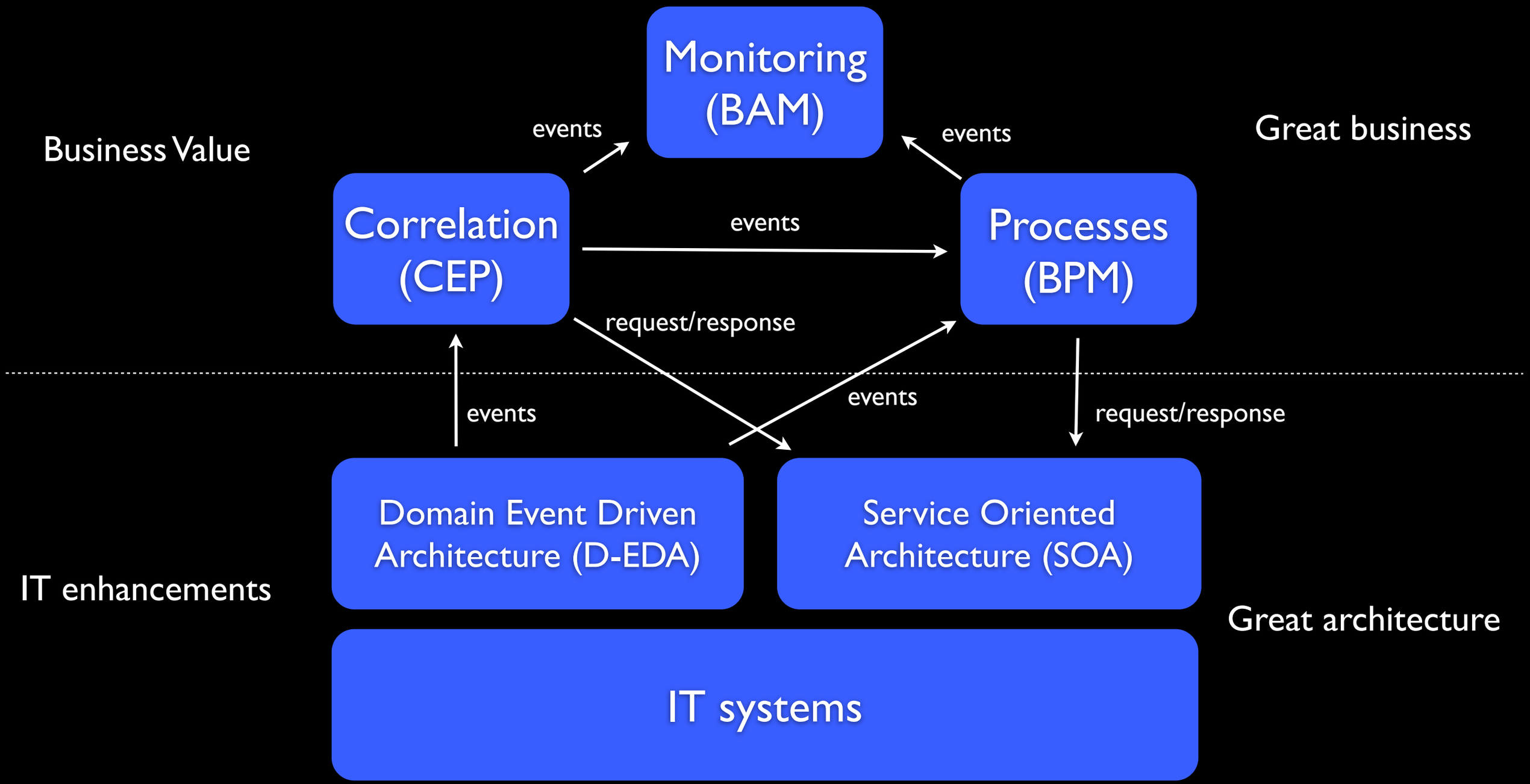

Customer System
Evaluate Highroller Start
Get Customer Info
Profit > €5000
Manual investigation / decision
Stop
decision?
Profit > €1000
Set VIP Status
Stop
Manager approval
approved?
Stop
PotentialHighrollerEvent

Business Activity Monitoring (BAM)

- Aggregation, analysis, and presentation of real time information about activities inside organizations and involving customers and partners
- BAM attempts to do for business processing what network management tools do for network operations

Figure 3 Oracle BAM dashboard supporting financial services scenario

The real-time business eco system



Thank you!