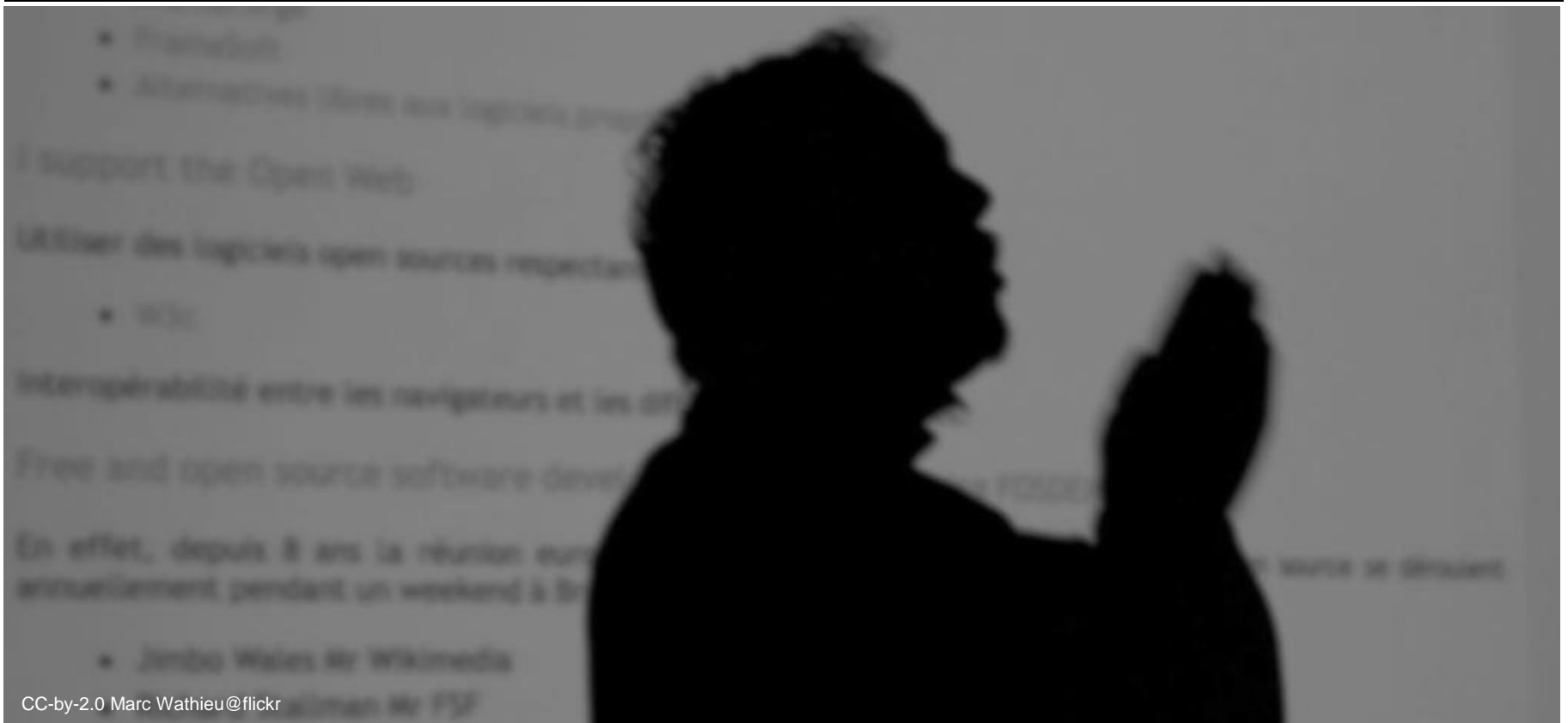


Discussion of Paper: Middleware – A Model for Distributed System Services



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Stefan Appel, Tobias Freudenreich, Sebastian Frischbier
A. Buchmann



CC-by-2.0 Marc Wathieu@flickr

- “Middleware: A Model for Distributed System Services”
- Definitions
 - Middleware: “Middleware can be described as "software glue".“ (Wikipedia)
 - Framework: “a universal, reusable software platform used to develop applications, products and solutions” (Wikipedia)
- Middleware: technical perspective?
- Framework: developer perspective?



Example

MESSAGING MIDDLEWARE

Message-oriented-Middleware (MOM)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Middleware for communication between components
- **Point-to-Point** and **Publish/Subscribe** Communication
 - Decoupled, Asynchronous, One-to-Many
- Standards
 - **API:** Java Message Service (JMS)
 - **Protocol:** Advanced Message Queuing Protocol (AMQP)
- Products:
 - JMS: Apache ActiveMQ, JBoss HornetQ, Oracle WebLogic, ...
 - AMQP: Apache Qpid, VMWare RabbitMQ, ...
 - Between 20 and 30 JMS Brokers, between 5 to 10 AMQP Brokers



JMS API - Sender

```
Connection jmsConnection = cf.createConnection();
```

```
Session jmsSession = jmsConnection.createSession(false,  
Session.AUTO_ACKNOWLEDGE);
```

```
...
```

```
TextMessage m = jmsSession.createTextMessage("This is payload of  
message");
```

```
producer.send(m);
```

4.6. *MessageProducer*

A client uses a `MessageProducer` to send messages to a `Destination`.
A `MessageProducer` is created by passing a `Queue` or `Topic` to a session's `createProducer` method.

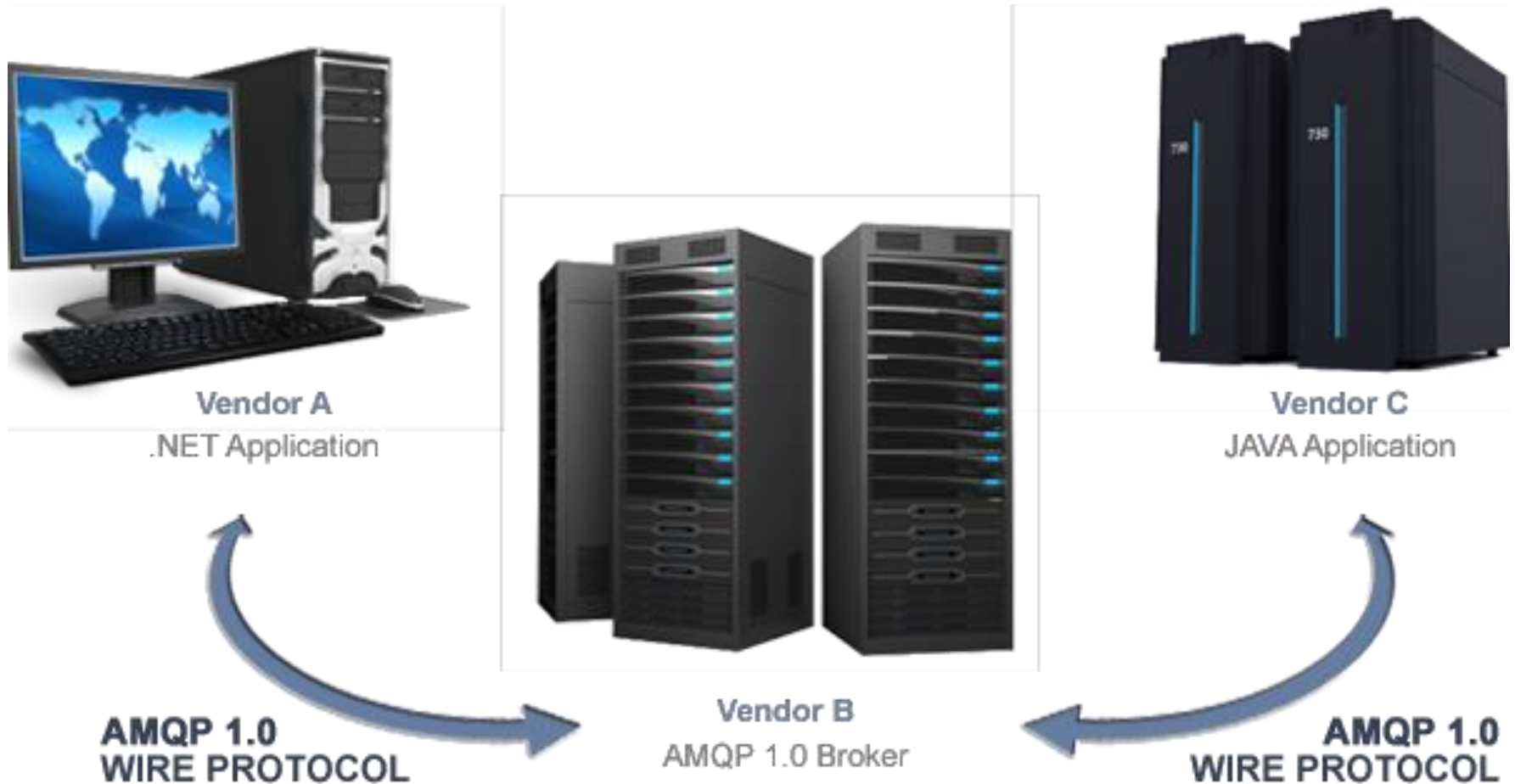
3.4.6. *JMSReplyTo*

The `JMSReplyTo` header field contains a `Destination` supplied by a client when a message is sent. It is the destination where a reply to the message should be sent.

AMQP enables Interoperability



TECHNISCHE
UNIVERSITÄT
DARMSTADT



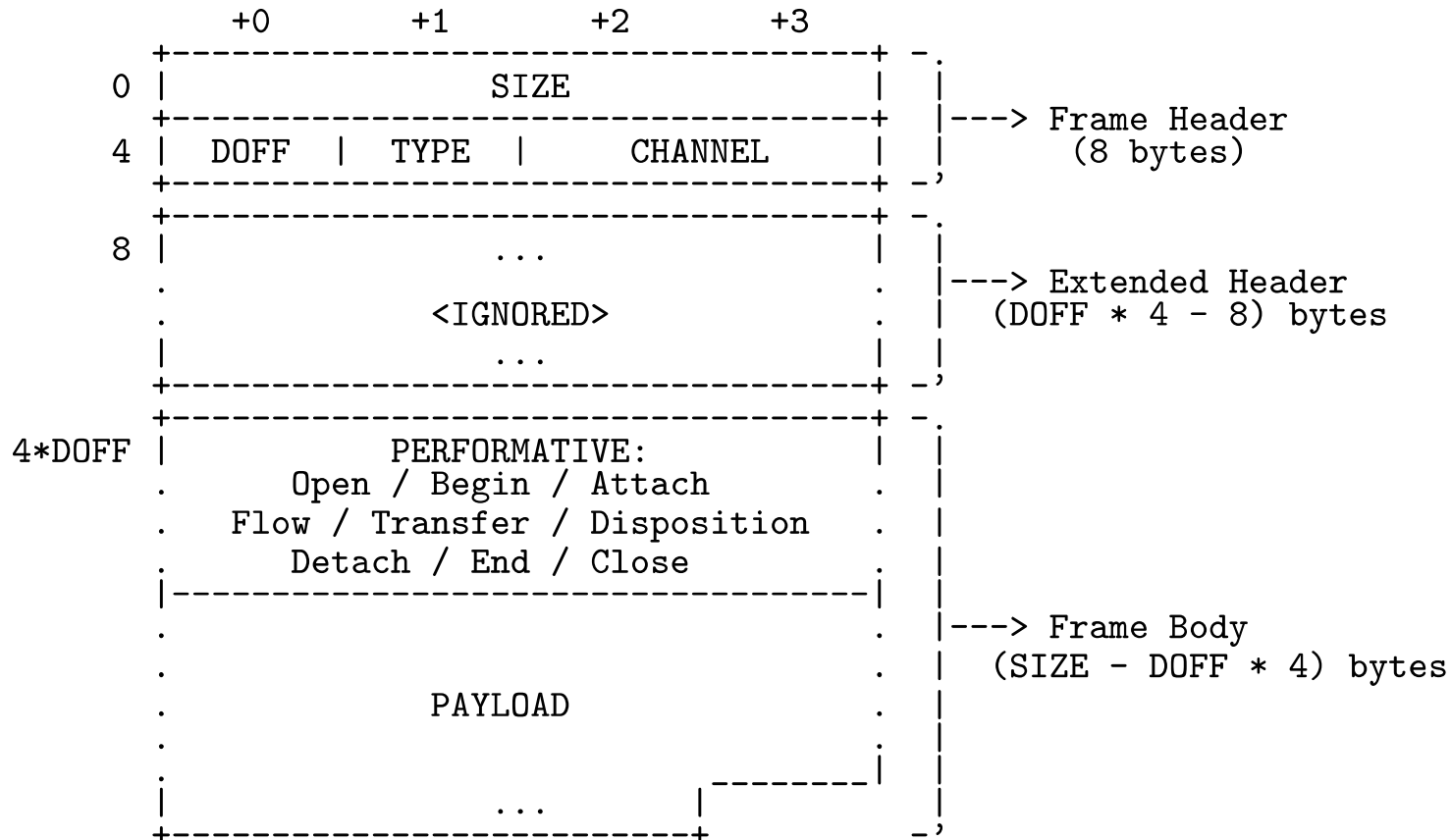
Source: <http://www.amqp.org>

AMQP Protocol Specification



TECHNISCHE
UNIVERSITÄT
DARMSTADT

type: 0x00 - AMQP frame



- Communication between message producer and consumer
- API vs. Protocol
 - What role does a protocol play?
 - What role does an API play?
- Vendor lock-in and interoperability
 - **API:** Code remains unchanged while middleware service provider can be changed. But bound to programming language?
 - **Protocol:** If you want to implement the protocol by yourself, interoperability is given. But what about libraries?
- What is the relation between API and Protocol?
 - 1:1, 1:n, n:1, m:n

Message-oriented Middleware

- Standardization
 - API: describe behavior?
 - Protocol: specify behavior?
- JMS – a Middleware Service
 - JMS brokers are stand-alone products
 - JMS also part of Java Application Servers (Java EE)
 - Is JMS a middleware service or a middleware?
- Data integration
 - Message-oriented middleware transports data from A to B
 - What about data semantics?
 - Managed by middleware?

Questions?

- Next exercise available online.