XMPP-Grid: XMPP Protocol Extensions for Use in MILE

http://tools.ietf.org/html/draft-appala-mile-xmpp-grid-00

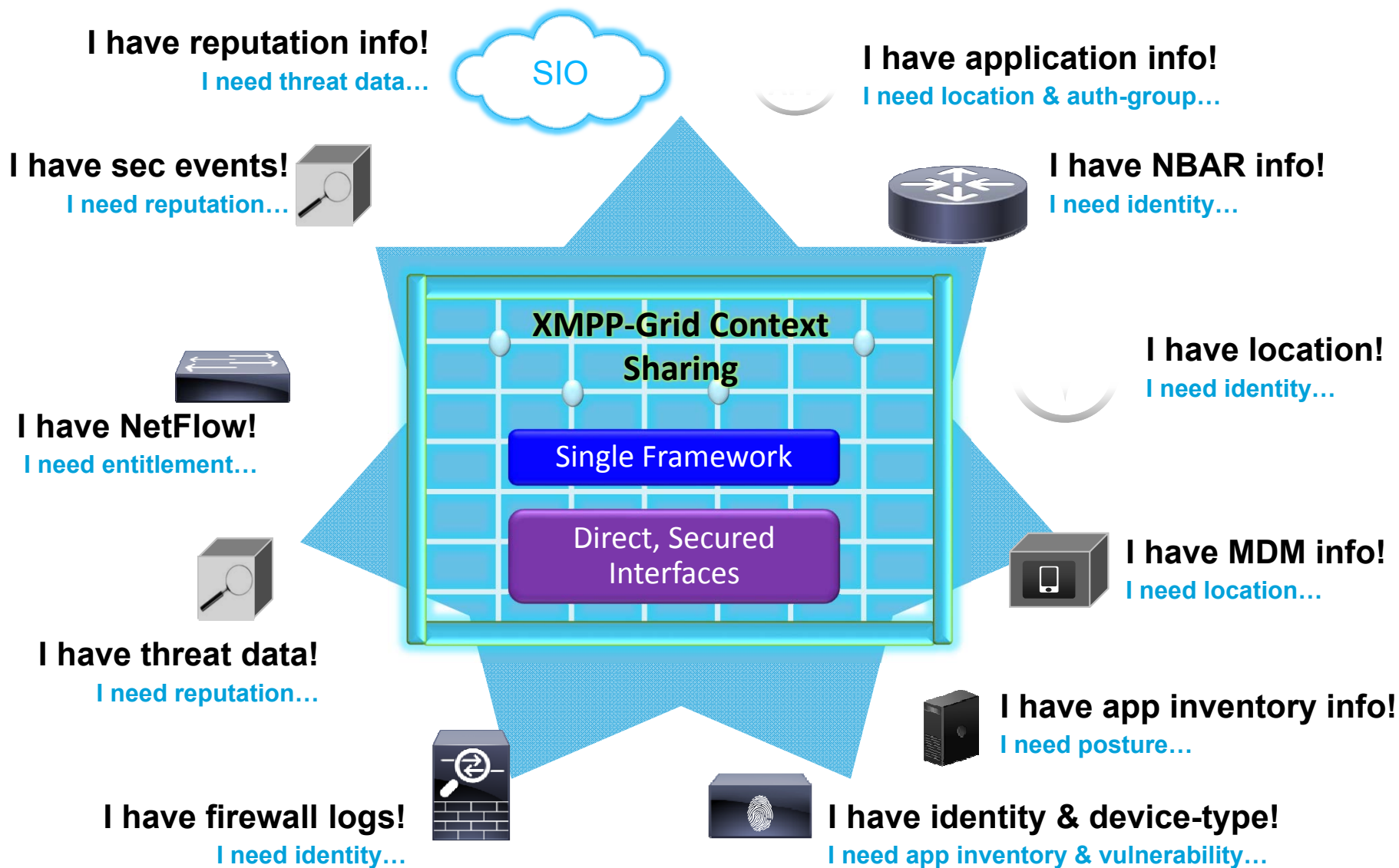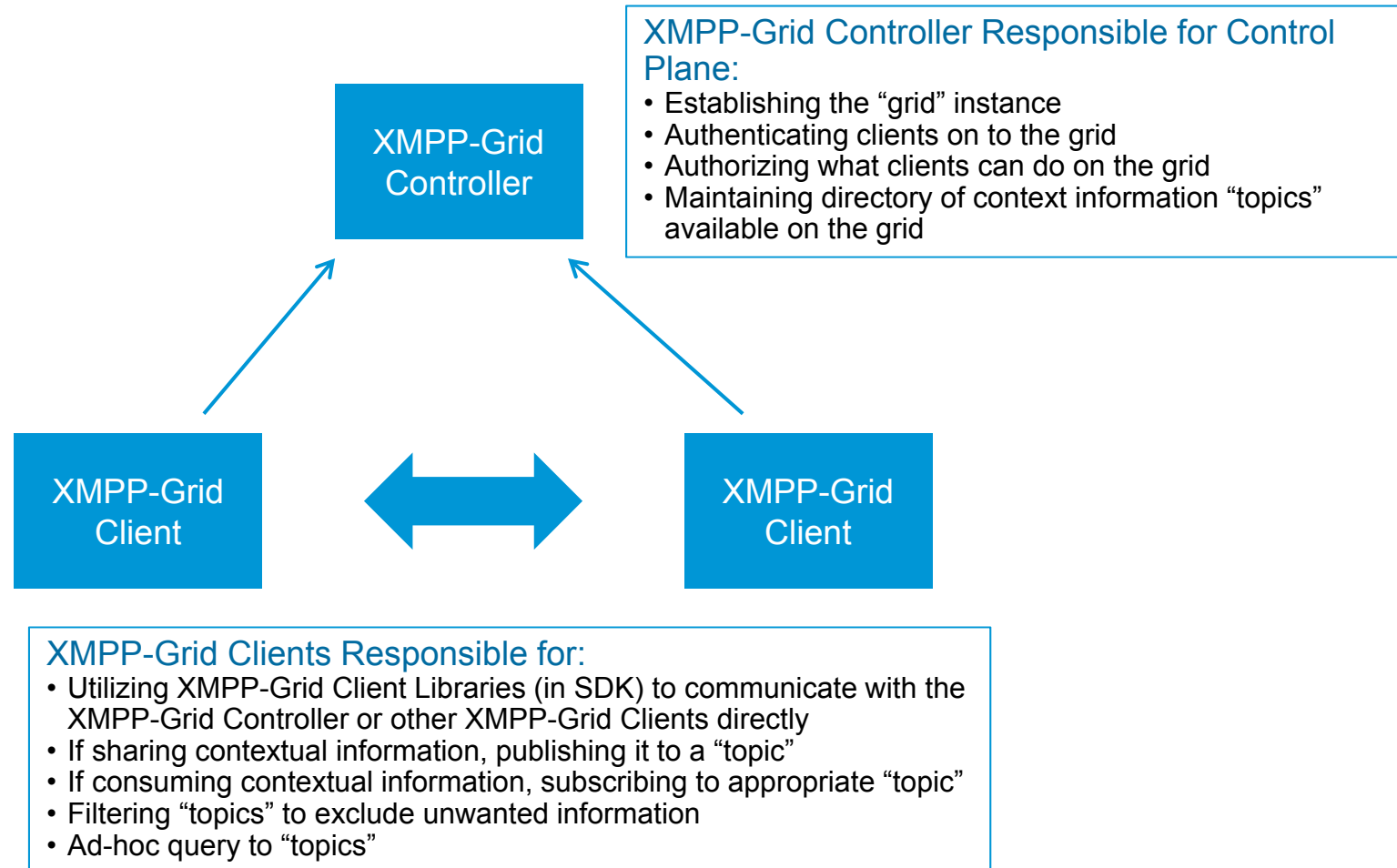Nancy Cam-Winget and
Syam Appala

November 2015

Agenda

- XMPP-Grid Use-Case

- Design Considerations

- XMPP as XMPP-Grid Transport

- XMPP-Grid Controller & Control, Data Flow Segregations

- Client Authentication & Authorization

- XMPP-Grid Protocol

- Topics & Subtopics with message filters

- IODEF Applicability

XMPP-Grid

Enabling the Potential of Network-Wide Information Sharing

I have reputation info!
I need threat data…

SIO

I have application info!
I need location & auth-group…

I have sec events!
I need reputation…

I have NBAR info!
I need identity…

XMPP-Grid Context Sharing

I have location!
I need identity…

I have NetFlow!
I need entitlement…

Single Framework

Direct, Secured Interfaces

I have MDM info!
I need location…

I have threat data!
I need reputation…

I have app inventory info!
I need posture…

I have firewall logs!
I need identity…

I have identity & device-type!
I need app inventory & vulnerability…

XMPP-Grid Architecture & Components

XMPP-Grid Controller

XMPP-Grid Client

XMPP-Grid Client

XMPP-Grid Controller Responsible for Control Plane:
- Establishing the "grid" instance
- Authenticating clients on to the grid
- Authorizing what clients can do on the grid
- Maintaining directory of context information "topics" available on the grid

XMPP-Grid Clients Responsible for:
- Utilizing XMPP-Grid Client Libraries (in SDK) to communicate with the XMPP-Grid Controller or other XMPP-Grid Clients directly
- If sharing contextual information, publishing it to a "topic"
- If consuming contextual information, subscribing to appropriate "topic"
- Filtering "topics" to exclude unwanted information
- Ad-hoc query to "topics"

XMPP-Grid Controller Design Goals

XMPP-Grid Server

Controller

Transport

- **Policy-based Authorization**

    Centralized control for authorization and client management

    Facilitates secure communication between authorized clients

- **Scalable**

    Architecture scales to thousands of clients/nodes

   Provide resilient, high availability support

- **Agile**

    Enable many different uses across the communication fabric i.e. context, policies …

    Should be platform agnostic (C/C++, Python, Java …)

    Negotiation for type of data plane communication & APIs

- **Lightweight Client**

    Enable adoption through small footprint & intuitive APIs
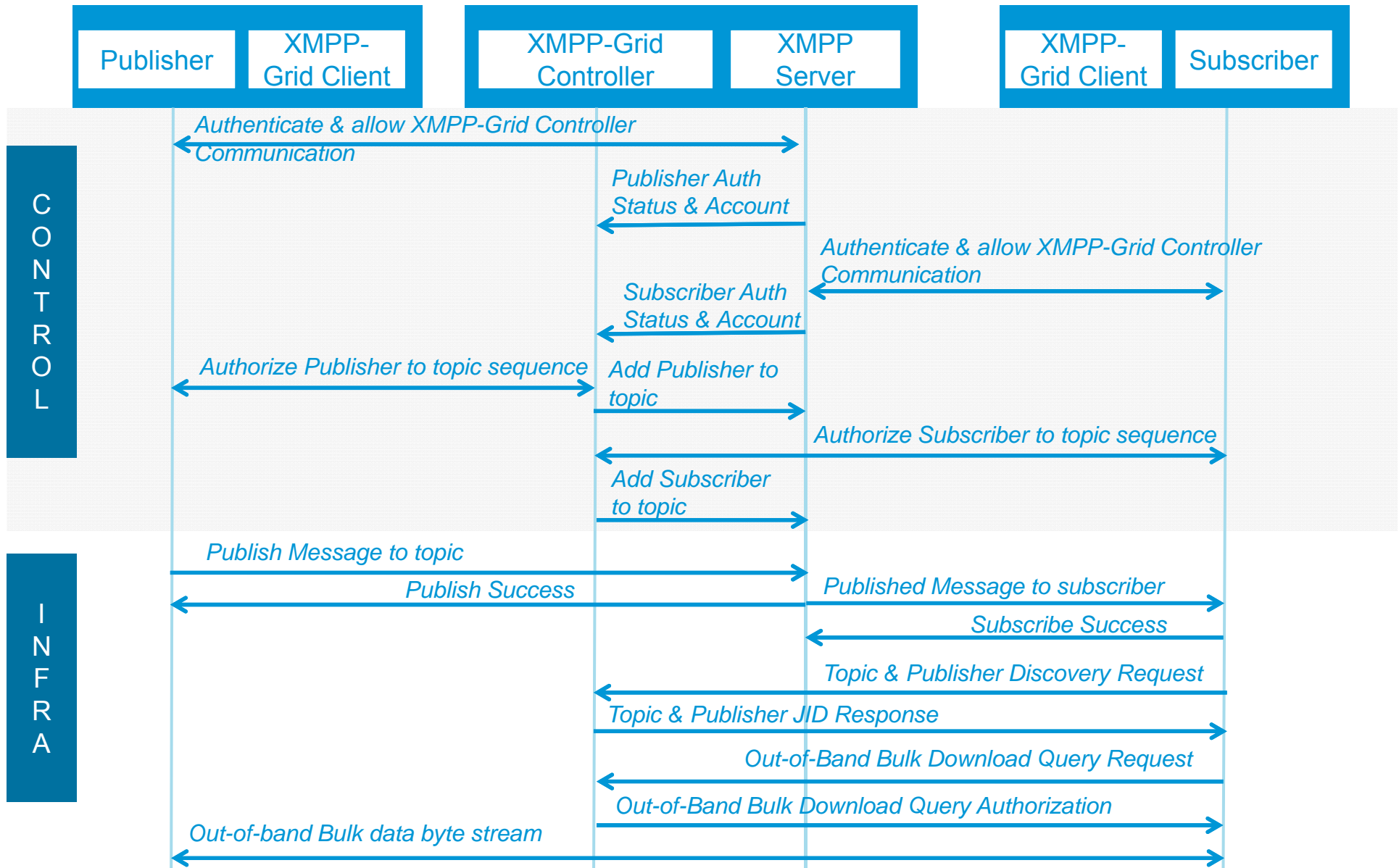
- **Standards**

    Enable adoption through standardization of schemas & information models
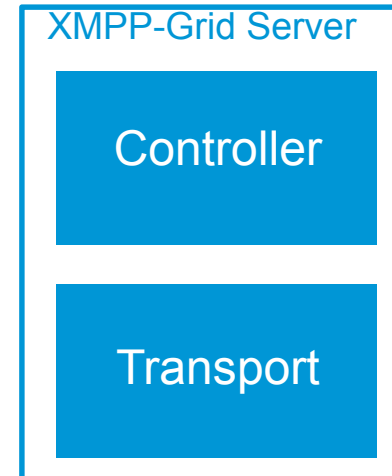
Why XMPP for XMPP-Grid Transport?

- **Open** – standards-based, decentralized (no single point of failure) and federated architecture

- **Real-time eventing** – using publish, subscribe notifications

- **Security** – Domain segregations; federation support; strong security via SASL and TLS

- **Flexibility** – Custom functionality can be built on top of XMPP; Easily extensible

- **Bi-directional** - avoids firewall tunneling

- **Scalable** – supports cluster mode deployment and message routing

- **Peer-to-peer** – directed queries and OOB file transfer support

- + Presence, service and device capability discovery …

XMPP-Grid Control & Data Flow

Publisher	XMPP-Grid Client		XMPP-Grid Controller	XMPP Server		XMPP-Grid Client	Subscriber

CONTROL

Authenticate & allow XMPP-Grid Controller Communication

Publisher Auth Status & Account

Authenticate & allow XMPP-Grid Controller Communication

Subscriber Auth Status & Account

Authorize Publisher to topic sequence

Add Publisher to topic

Authorize Subscriber to topic sequence

Add Subscriber to topic

INFRA

Publish Message to topic

Publish Success

Published Message to subscriber

Subscribe Success

Topic & Publisher Discovery Request

Topic & Publisher JID Response

Out-of-Band Bulk Download Query Request

Out-of-Band Bulk Download Query Authorization

Out-of-band Bulk data byte stream

XMPP-Grid Client Registration

3 layer security model –
Mutual-cert based authentication + account approval + authorization group assignment with policy control

- Auto registration

  Clients that server can authenticate (using x.509 certificates) will have their accounts auto created after authentication

  Clients can specify authorization group of interest

- Manual registration

  Administrator has to approve/decline client accounts after their authentication

  Administrator can assign authorization group to the client resulting in client logoff and logging back in for the group change to take effect
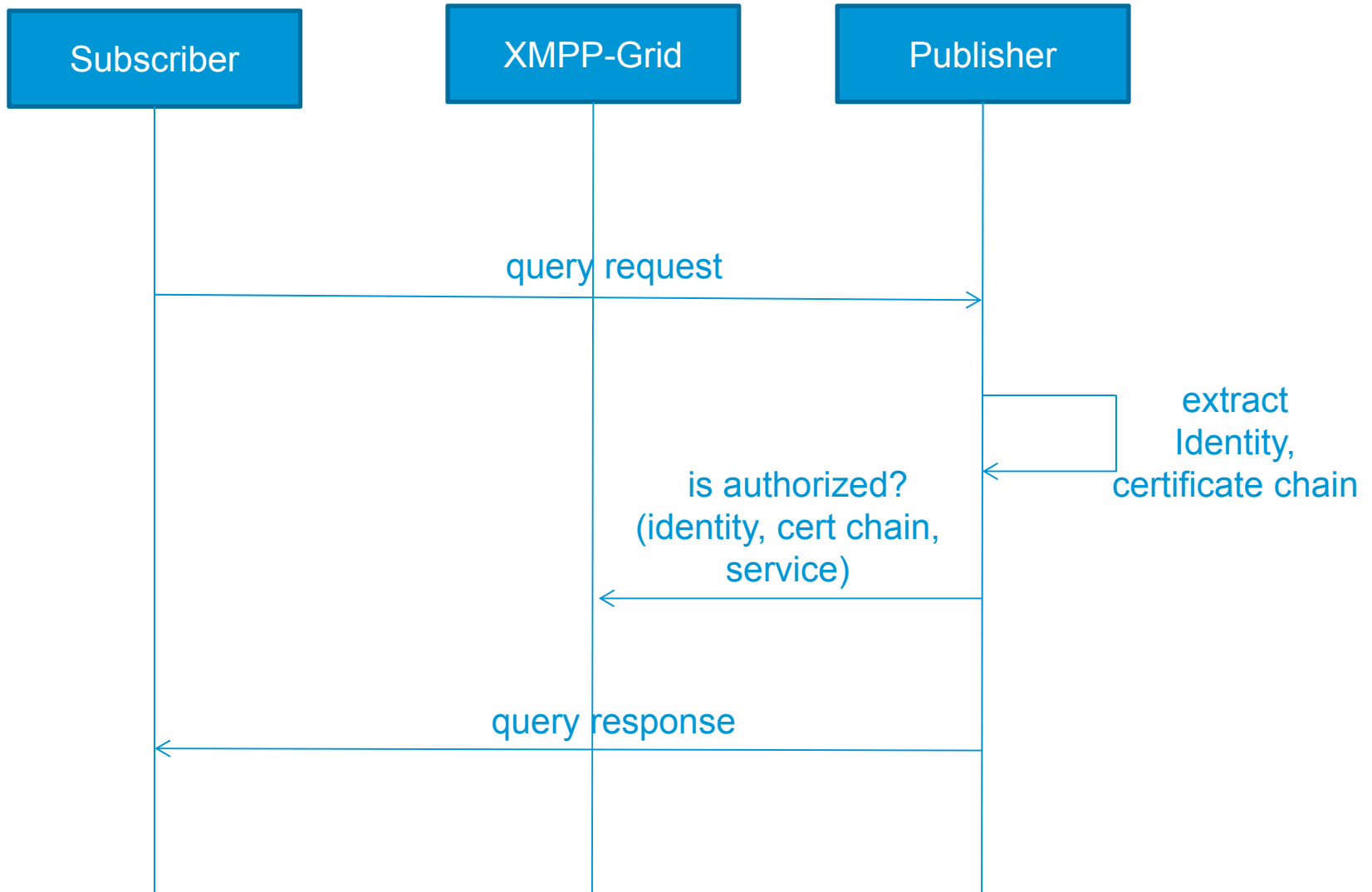
Client Registration
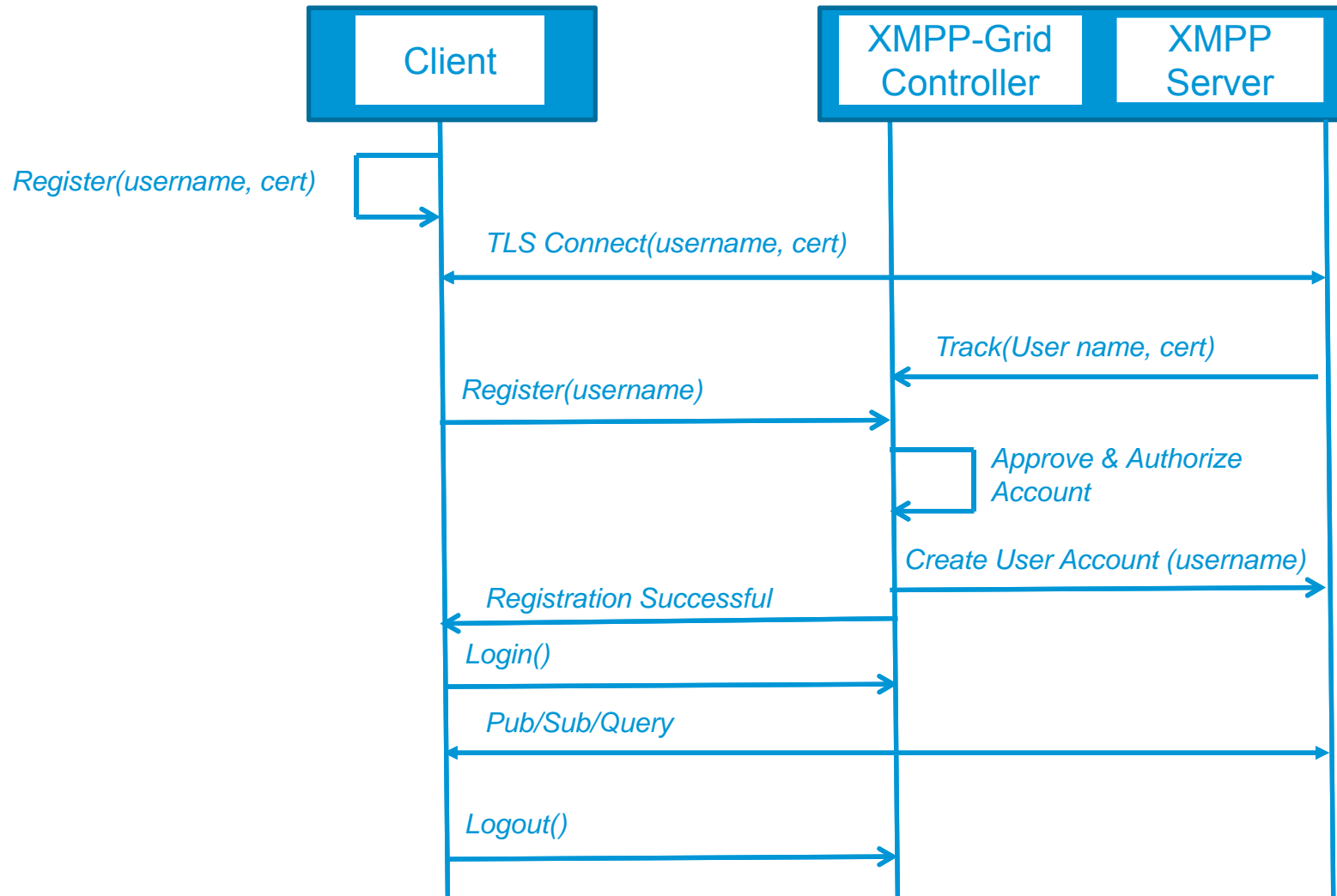
XMPP-Grid Client Authorization

- Authorization policies can be based on attributes such as

   Authorization group, Topic name, client name, device type, operation …

- Controller authorizes clients to publish or subscribe to a topic at "subscribe" time

- Publisher, when it receives a directed (peer-to-peer) or bulk download query from a subscriber, asks the controller for authorization using XMPP-Grid client identity

Publish/Subscribe Authorization

Publisher/Subsc riber

XMPP-Grid Controller

XMPP Server

Publish or Subscribe

extract identity

is authorized? (identity, publish/ subscribe)

Publish or Subscribe
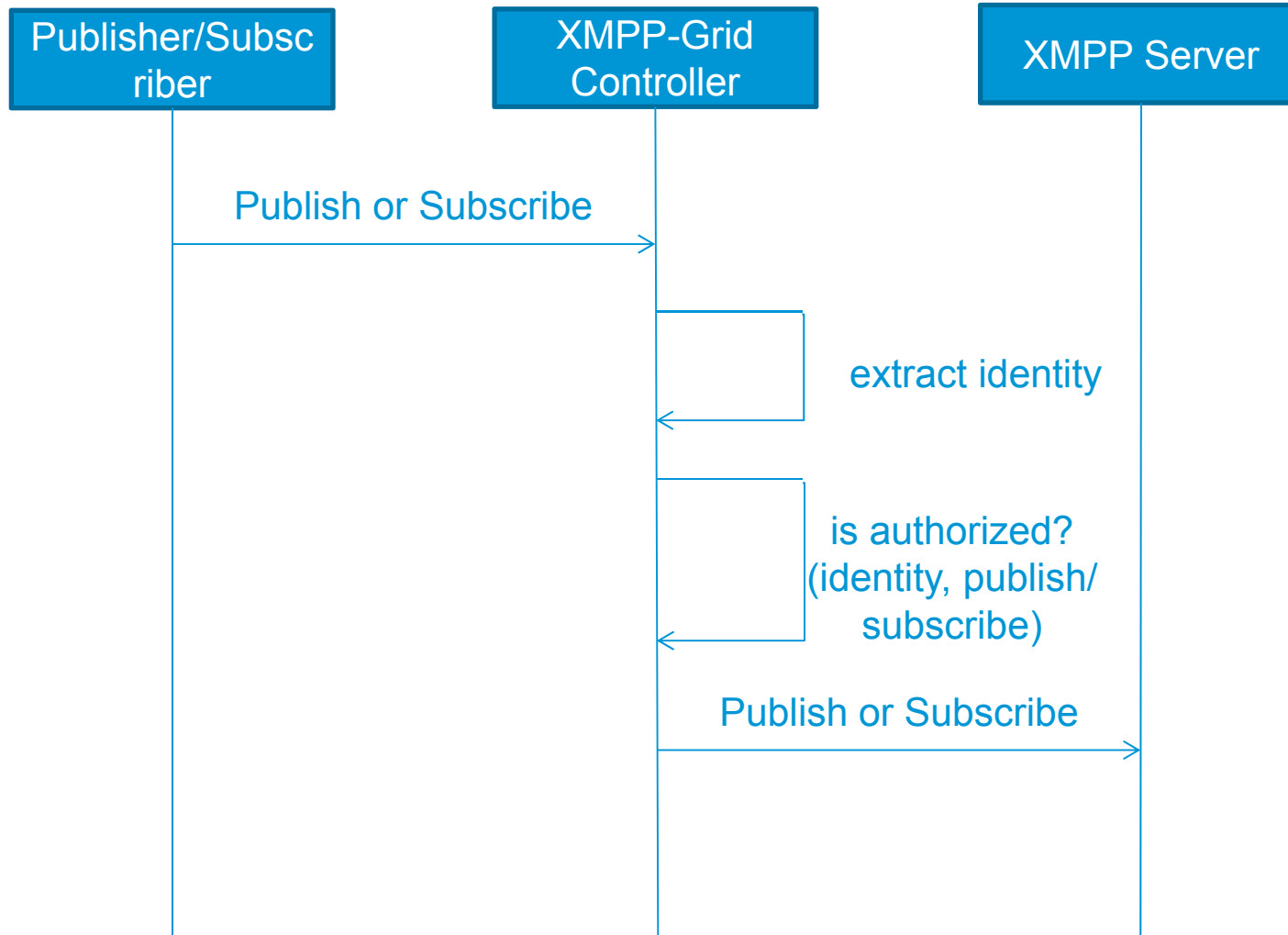
Directed/Bulk Query Authorization

query request

extract
Identity,
certificate chain

is authorized?
(identity, cert chain,
service)

query response

XMPP-Grid Topics through Capability

- Capability providers:
  - publishes information with a defined schema on XMPP topic(s)
  - defines XML schema, topic version, available queries and notifications for each topic
  - publishes the messages to one or more XMPP topics depending on:

    Mutually exclusive schemas – create one topic per schema

    Same schema, but subscribers desire only a subset of attributes and values – XMPP-Grid creates subtopics and uses message filters to deliver filtered information

- Topics are discoverable on XMPP-Grid through XMPP-Grid protocol query

IODEF as an XMPP-Grid Topic

- IODEF as a common data format can be expressed as an XMPP-Grid Topic: RFC 5070 defines the XML schema

- XEP-0268 defines the means for XMPP deployments to use IODEF reports

Backup

15

XMPP-Grid Protocol

- Infrastructure protocol that enables client application to be agnostic to data plane protocol, XMPP

- Makes use of the XMPP transport and introduces an application layer protocol leveraging XML and XMPP extensions to define the protocol

- Provides interfaces for

    Register, login, logout

    Query to discover topics, capability provider discovery, directed peer-to-peer

    Register as a publisher or subscriber to topic (information channel with publishers and subscribers sharing a well defined publisher data model)

- XMPP-Grid clients connect to the XMPP-Grid using the XMPP-Grid Protocol

- Capability providers extend the XMPP-Grid Protocol infrastructure model and define capability specific models, allowing a cleaner separation of infrastructure and capabilities that can run on XMPP-Grid

XMPP-Grid addresses …

- Visibility into *"who is connecting", "who is accessing what"*

- Centralized, policy-based authorization – *"who can do what"*

- Secure, bidirectional connectivity

- Mutual certs-based authentication

- Flexible consumption APIs – real-time, on-demand, bulk transfer

- Client contextual needs support through semantic, syntactic filtering

- Ability for peers to negotiate out-of-band, secure p2p connection

- Standardize schemas & information models through XML

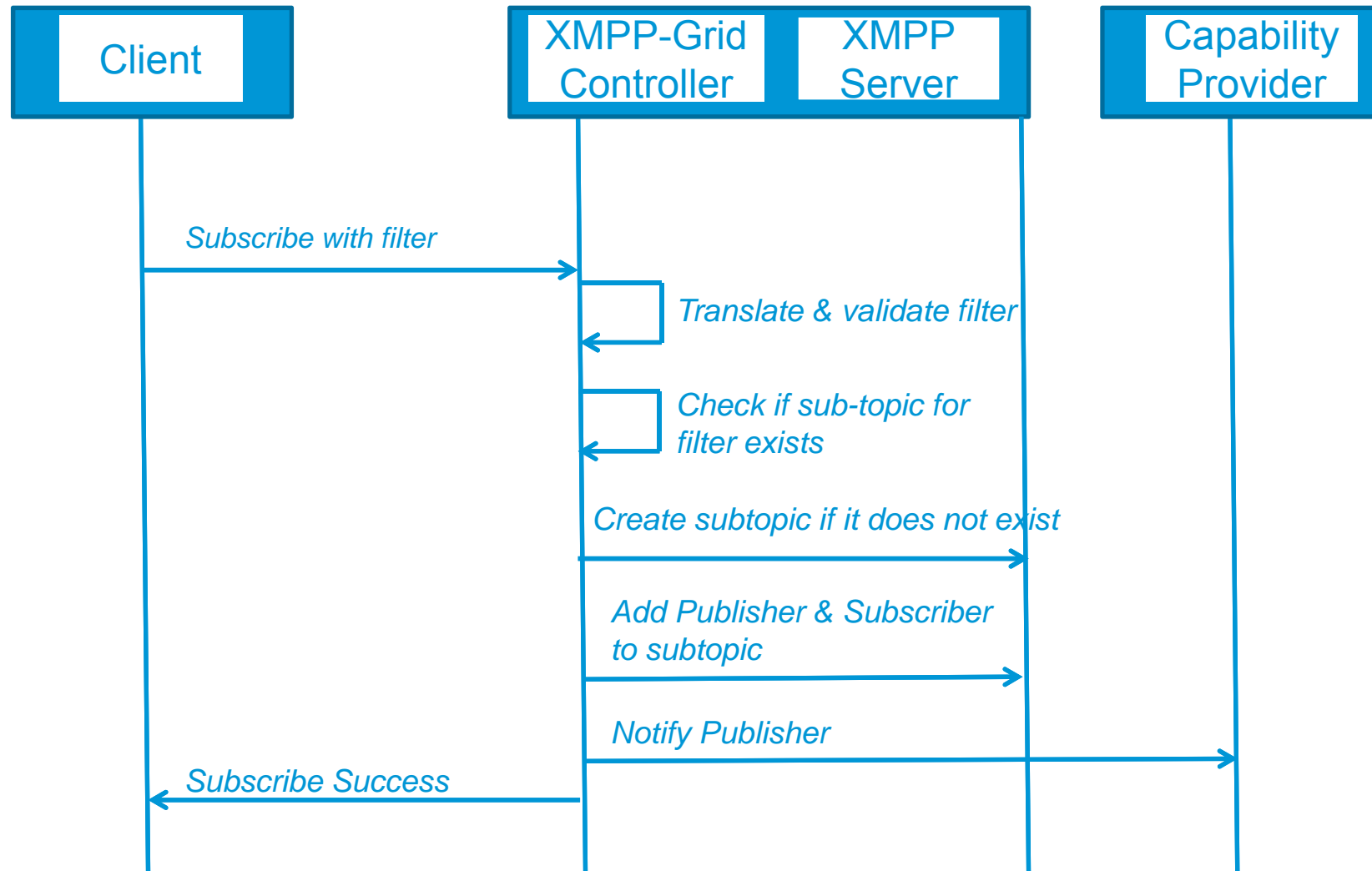- Scalable to thousands of nodes

- Platform agnostic

XMPP-Grid Controller

- Plugs-in as external component to the XMPP server

- Responsible for –

  Account approvals of XMPP-Grid clients

  Authorization of client actions – subscribe, publish, query, bulk download

  Topic (information channel with publishers and subscribers sharing a well defined publisher data model) setup with subscription list

  Maintains directory of topics & topic subscriptions

  Communicates with other XMPP-Grid controller in cluster for HA

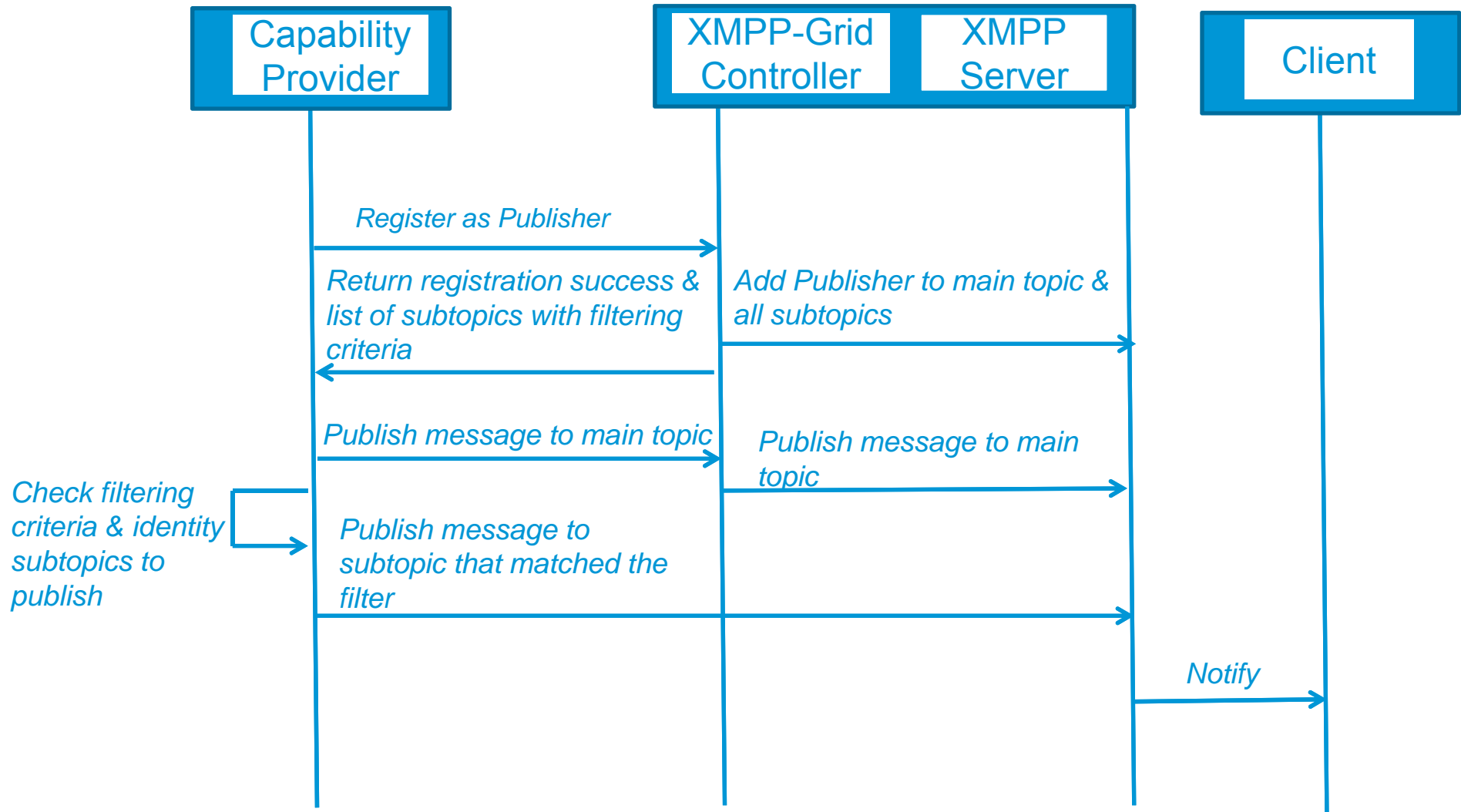 Offers interfaces & statistics for management of clients & topics

XMPP-Grid Client Authentication

- Each XMPP-Grid client will go through the phases of authentication, registration and authorized access

- Certs-based mutual authentication between client and server using X.509 certificates

- Mutual authentication and tunnel establishment through XMPP "SASL External"

- If client certificate passes validation client registration requests are relayed only to XMPP-Grid controller for account approval

- If client certificate does not pass validation, the connection is terminated with XMPP standards-based error messages

Subtopic Creation Flow

Client | XMPP-Grid Controller | XMPP Server | Capability Provider

Subscribe with filter

Translate & validate filter

Check if sub-topic for filter exists

Create subtopic if it does not exist

Add Publisher & Subscriber to subtopic

Notify Publisher

Subscribe Success

Publish on Subtopics Flow

Capability Provider

XMPP-Grid Controller

XMPP Server

Client

Register as Publisher

Return registration success & list of subtopics with filtering criteria

Add Publisher to main topic & all subtopics

Publish message to main topic

Publish message to main topic

Check filtering criteria & identity subtopics to publish

Publish message to subtopic that matched the filter

Notify

XMPP-Grid Protocol Example

```
// Capability Provider Discovery Request

<iq id="996IL-8" to="grid_controller.jabber" from="asa@syam-06.domain.com/syam-mac" type="get">

  <grid xmlns='gi' type='request'>

   <DiscoveryQuery xmlns='com.domain.gi.gcl.controller'>

    <find><param xsi:type="xs:string" xmlns:ns2="gi" xmlns:xs=" xmlns:xsi=">com.domain.ise.session.SessionQuery</param></find>

   </DiscoveryQuery>

  </grid>

</iq>



// Capability Provider Discovery Response

<iq from='grid_controller.jabber' id='996IL-8' to='asa@syam-06.domain.com/syam-mac' type='result' xmlns='jabber:client'>

  <grid type='response' xmlns='gi'>

   <DiscoveryQuery xmlns='com.domain.gi.gcl.controller'>

    <find xmlns=""><value xmlns:ns3='http://jaxb.dev.java.net/array' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:type='ns3:stringArray'><item>ise@syam-06.domain.com/syam-mac</item></value></find>

   </DiscoveryQuery>

  </grid>

</iq>
```

XMPP-Grid Subtopics & Message Filters

- Capability provider specifies semantic filters such as location, domain etc it supports for a given topic at subscribe time to the controller

- Subscribers discover the topics & supported message filters, and specify filters of interest to them to the controller

- Controller groups subscribers based on the expressed message filters, creates subtopics under the main topic and notifies the Publisher about the created subtopic

- Publisher publishes a message on the main topic and on the subtopics, after applying the message filter

Subtopics & Message Filters

- Controller cleans up the subtopics if subscription list is 0, to avoid proliferation of subtopics

- Pub/Sub, directed and bulk query can be supported for subtopics also – it all depends on the capability provider

- Message filters can be applied on XMPP-Grid server side instead –instead of publishing on subtopic, capability provider publishes on main topic and XMPP-Grid Pub/Sub component can apply filter messages

  Server-side message filters and specific message filter mechanisms such as XPATH are beyond the scope of this specification