Resource-Oriented Lightweight Information Exchange (ROLIE)

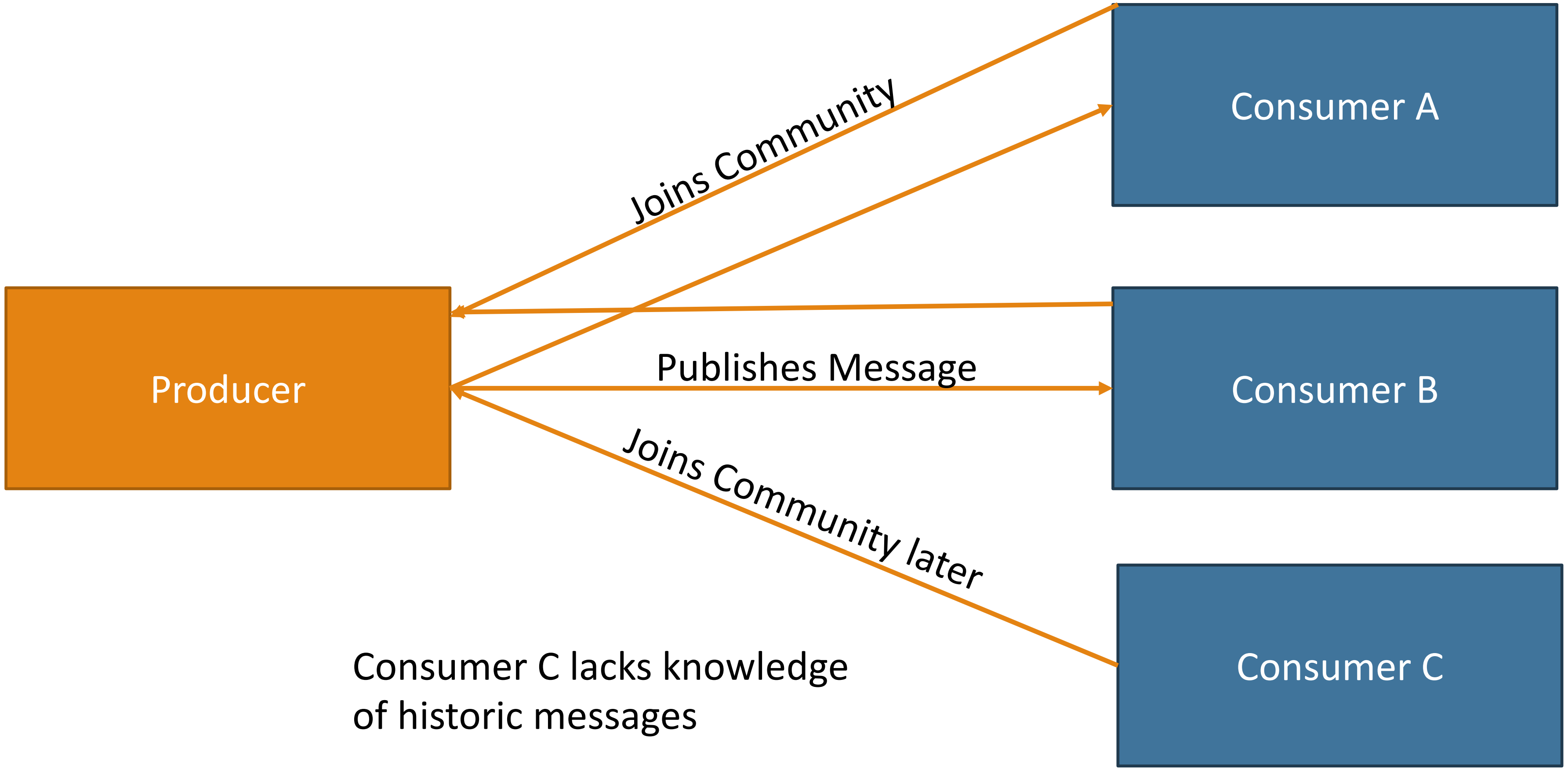
draft-ietf-mile-rolie-05

IETF 97 - SACM Working Group
Dave Waltermire, Stephen Banghart

The Problem

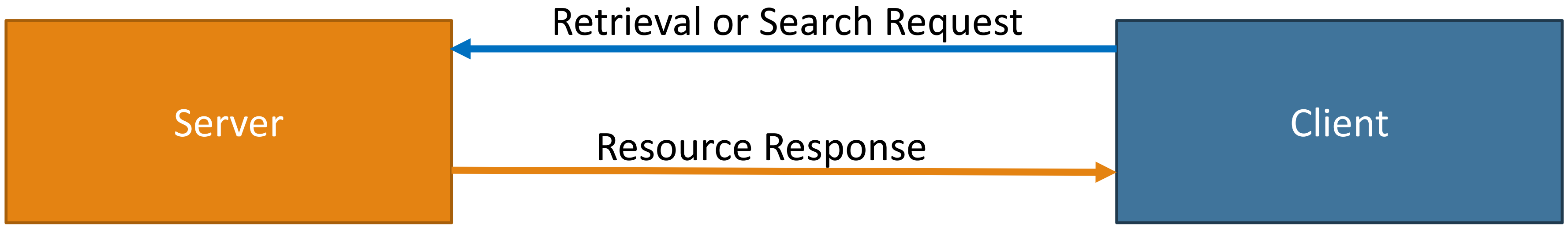
- Point-to-point communication requires high degree of coordination.
- This degree of coordination makes wide scale public distribution difficult
- Point-to-point communications are more difficult to automate
- Organizations require security information from many sources

Message-Oriented Publish/Subscribe Model



Consumer C lacks knowledge
of historic messages

Message-Oriented Request/Response Model



What if the resource or search term is unknown?

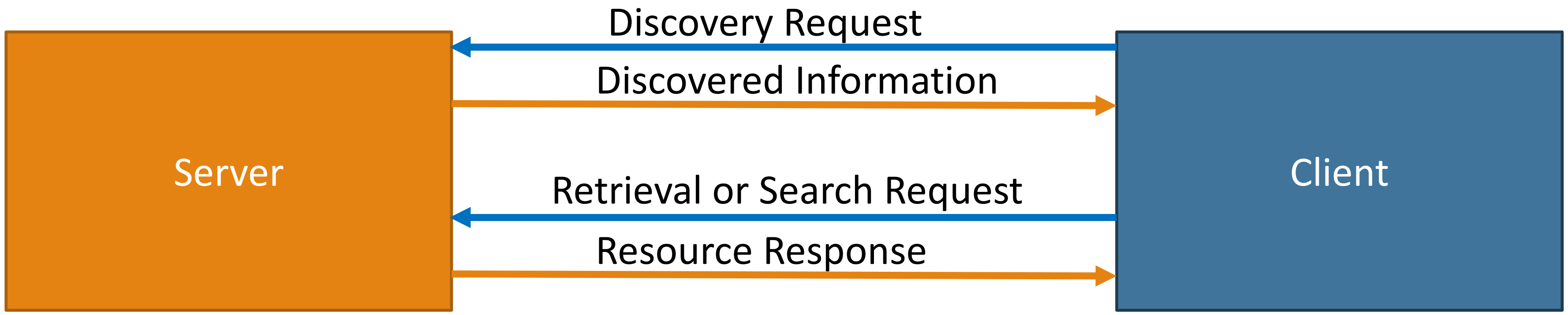
The Goals of ROLIE

- Provide metadata to allow clients to discover and search information resources
- Minimize the retrieval of unneeded information and reduce round trips
- Provide granular control of access to resources in order to support public and private information exchange
- Facilitate exchange of information without requiring pre-established sharing consortiums
- Enable automatic machine communication and information processing

ROLIE as a Solution

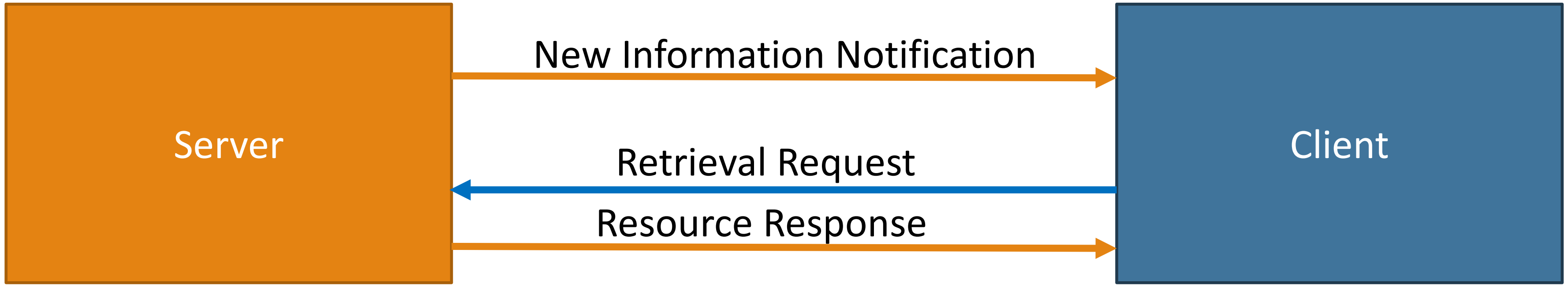
- The Resource-Oriented Lightweight Resource Exchange (ROLIE) is a profile of the Atom Publication Protocol and the Atom Syndication Format.
- Allows collections of security information resources to be discovered without prior knowledge of the information.
- Provides a mechanism to characterize different types of security information resources
- Creates system for producers to push content with granular access controls
- Originally meant for IODEF exchange, repurposed as a general security information exchange

Resource-Oriented Discovery Model



Available resources can be discovered, and interesting resources can be searched and retrieved.

Resource-Oriented Publication/Subscription



Publication/Subscription protocol (e.g., XMPP) can be used with the Resource-Oriented approach to provide notifications of new information.

Anatomy of a ROLIE Service Document

```
<?xml version="1.0" encoding="UTF-8"?>
<service xmlns="http://www.w3.org/2007/app"
    xmlns:atom="http://www.w3.org/2005/Atom">
  <app:workspace>
    <atom:title>Public Security Information Sharing</atom:title>
    <app:collection
        href="http://example.org/provider/vulns">
      <atom:title>Public Vulnerabilities</atom:title>
      <app:categories fixed="yes">
        <atom:category
            scheme="urn:ietf:params:rolie:information-type"
            term="vulnerability"/>
      </app:categories>
    </app:collection>
  </app:workspace>
</app:service>
```

Defines the type of information contained within a collection

Anatomy of a ROLIE Feed

```
<?xml version="1.0" encoding="UTF-8"?>
<atom:feed xmlns="http://www.w3.org/2005/Atom">
    <atom:id>http://example.org/provider/vulns</atom:id>
    <atom:title>Public Vulnerabilities</atom:title>
    <atom:category scheme="urn:ietf:params:rolie:information-type"
        term="vulnerability" />
    <atom:updated>2012-08-05T18:13:51Z</atom:updated>
    <atom:link rel="self"
        href="http://example.org/provider/vulns" />
    <atom:link rel="service"
        href="http://example.org/rolie/servicedocument" />
    <atom:entry>
        ...
    </atom:entry>
</atom:feed>
```

Defines the type of information contained within a collection. Same as defined in the service document

Points to the service document associated with this feed.

Anatomy of a Paged ROLIE Feed

```
<?xml version="1.0" encoding="UTF-8"?>
<atom:feed xmlns="http://www.w3.org/2005/Atom">
    ...
    <atom:link rel="self" href="example.org/provider/vulns?page=5"/>
    <atom:link rel="first" href="example.org/provider/vulns?page=1"/>
    <atom:link rel="prev" href="example.org/provider/vulns?page=4"/>
    <atom:link rel="next" href="example.org/provider/vulns?page=6"/>
    <atom:link rel="last" href="example.org/provider/vulns?page=10"/>
    ...
</atom:feed>
```

Provides link relations for navigation through paged feed entries.

Anatomy of a ROLIE Entry

```
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns="http://www.w3.org/2005/Atom"
    xmlns:rolie="urn:ietf:params:xml:ns:rolie-1.0">
  <atom:id>http://www.example.org/provider/vulns/123456</id>
  <atom:title>Sample Vulnerability</title>
  <atom:updated>2012-08-05T18:13:51Z</updated>
  <rolie:format ns="urn:ietf:params:xml:ns:iodef-2.0"/>
  <atom:content type="application/xml"
      src="http://www.example.org/provider/vulns/123456/data"/>
</atom:entry>
```

Provides information about the data model of the content.

Content is linked to, not embedded.

The ROLIE extension system

- The Atom Category element provides a flexible extension point for characterizing information.
- Information types are expressed using these category elements.

- List of officially supported information types registered in IANA table.
- New information types can be added over time

- Example: “ROLIE CSIRT Extension” document creates IANA table entries for:
  - Information Types: Incidents, Indicators

ROLIE Software Descriptor Extension

- -00 draft recently published, more work is needed

- Establishes the “software descriptor” information type
  - Identifies and characterizes software, software installers/packages, and software patches.
  - Software descriptors do not characterize installation records, running software, or configuration state
- A “software descriptor” is similar in concept to the SWIMA software data model
  - ROLIE repositories represent an ideal source for this information
  - SWIMA can be used when a repository doesn’t exist

- Lists ISO 2015 SWID Tags as a data format that expresses this information type

Looking Forward

- ROLIE draft at revision 5, currently in WGLC in MILE, more review needed!
- Review and adoption of the ROLIE Software Descriptor Extension by SACM
- Additional ROLIE extensions can be created (e.g., configuration setting checklists, vulnerability records/bulletins)
- We are willing to help anyone who would like to work on these extensions.

Discussion