System and Network Engineering

Toward a Dynamic Trust Establishment Approach for Multi-provider Intercloud Environment

Canh Ngo, Yuri Demchenko
{t.c.ngo, y.demchenko}@uva.nl

System and Network Engineering Group
University of Amsterdam

UNIVERSITY OF AMSTERDAM

Agenda

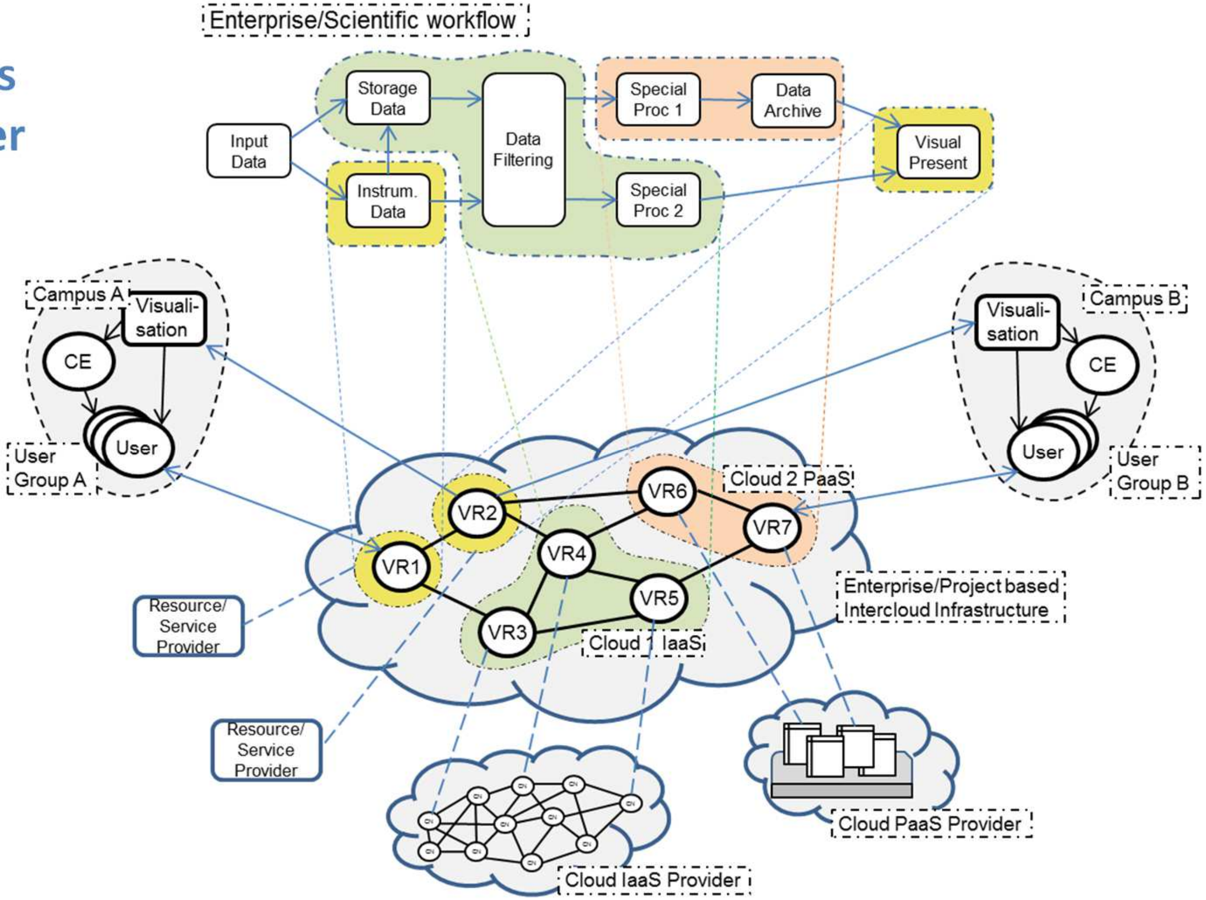
- Motivation
- Trust Management Challenges
- Trust Model
  - Attribute-based Trust approach
- Application
  - Dynamic Trust Establishment for Intercloud
  - Trust Evaluation Engine
- Conclusion and Future work

Motivation

System and Network Engineering

Intercloud use-cases

- **Enterprise IT infrastructure migration**
- **Large project-oriented scientific infrastructures**
- **IT infrastructure disaster recovery**



Universiteit van Amsterdam

Intercloud Properties

- Communication between Cloud providers/applications
  - Vertical integration: different service layers
  - Heterogeneous: cross-domains, composite services
- Distributed, public data access environment
- Data/resources are off-premise
- RORA*: cloud resource ownerships
  - Physical ownership
  - Management/brokering ownership
  - Subscription/consumption ownership

*RORA: Resource, Ownership, Role, Action (GEYSERS project)

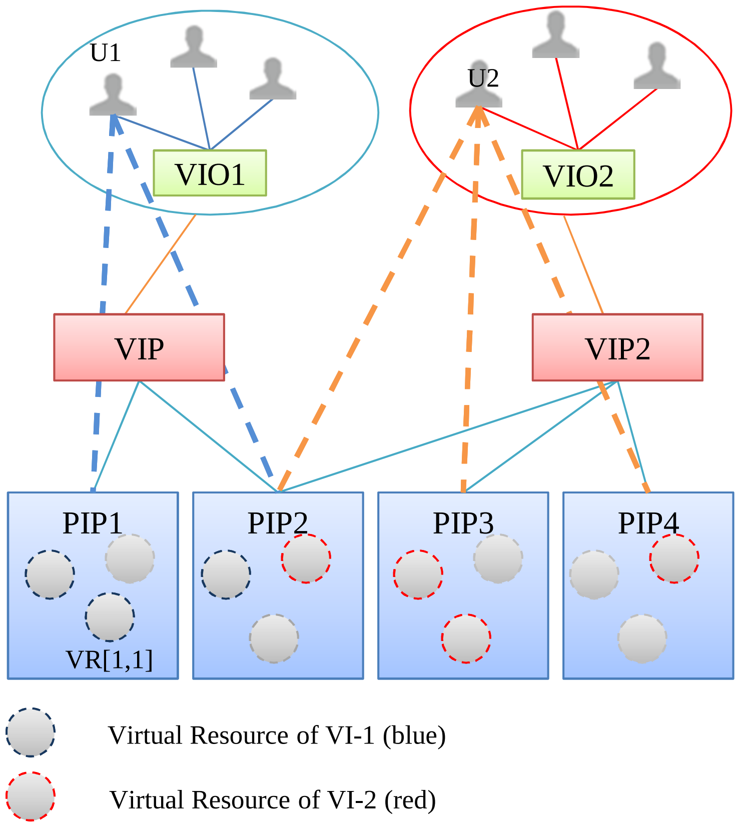
Challenges

- Distributed multiple security domains
  - Authorizations based on identities are not applicable
  - Attributed-based access control (ABAC): different attributes profiles at domains
- Clouds composed from multiple providers
  - Authorization for “unknown” entities (“know implicitly”)?
  - Relations between Cloud providers: dynamic, established on Cloud provisioning lifecycles
- Approach: Trust Management for distributed, public environment
  - Attribute-based, attribute semantics can be transformed between domains
  - Multiple levels of delegations
  - Dynamic trust-chain establishment
  - Efficient attribute-based trust evaluation implementation

Trust Model

- **Entities**
  - Cloud Providers
    - Physical Cloud Providers: PIP
    - Intermediate Cloud Providers: VIP, Cloud Broker
  - Cloud Clients
  - End-users/applications
- **Trust**

  "the belief of trustor in trustee to behave reliably, securely in a specific context"
- **Trust relationships**
  - **Properties**:
    - Asymmetric
    - Contextual
    - Time-constraint
  - **Types**:
    - Direct trust relationships
    - Indirect trust relationships


U1
U2
VIO1
VIO2
VIP
VIP2
PIP1
PIP2
PIP3
PIP4
VR[1,1]
Virtual Resource of VI-1 (blue)
Virtual Resource of VI-2 (red)

Trust Mechanisms(1)

- **Trust decisions**
  - Simple: binary (trust, distrust)
  - Complex: trust predicates
- **Attribute-based trust policies**
  - Attributes to describe trust context
  - Policy actor, policy target, policy context
  - Formal logic formula:

$$X= (x_1,x_2,\ldots x_n);\ x_i \in P_i$$

$$f(X) = \bigwedge_i \left[\bigvee_j \left(\bigwedge_k m_k \right) \right]$$

Trust Mechanisms(2)

- **Direct trust relationships**
  - Attributes:

$$X= (x_1,x_2,\ldots x_n);\ x_i \in P_i$$

  - Attribute-based trust policy:

$$f_{actor}(target, X) \rightarrow pred$$

- Actor, target: entities
- X: attribute-based context
- pred: predicates (e.g. trust, distrust, etc)

Trust Mechanisms(3): Delegation

- Indirect trust relationship?
- Delegation

  "Transferring part of the ownership (i.e., right to control as defined by the policy/administrative context) from the trustor to the trustee"

- Trust credential issuer policy

$$f_{trustor\_B}\,(trustee\_A, X) \rightarrow tc_B^X$$

tc: trust credential:
{trustor, trustee, context}

- Delegation policy

$$f_{trustor}^{d}(X) \rightarrow \{targets\}$$

X – trust context
d – abbrev. for delegation
targets – Id/trust_anchors of recommenders (e.g. B)

Trust Mechanisms(4): Delegation

- Example:

  “B delegates A to access (r,w, etc) cloud resource X at C”

- At A: access context description X
- At B: $f_B\,(A, X) \rightarrow tc_B^{X_A}$
- At C:
  - Delegation policy at *C* for context *X*

    $$f_C^d(X) \rightarrow targets := \{B\}$$

  - Trust policy for unknown entities

    $$f_C\,(?\,, X) := \left[X.\,tc_B^A : B \in f_C^d(X)\right] \rightarrow trust|pred$$

Trust Management: Challenges & Directions

- **Trust policy evaluation**: attribute-based policy evaluation
  - XACML with extensions
  - Using **Multi-data types Interval Decision Diagrams** (MIDD): neutralized with policy languages.
  - Efficient in evaluation complexity.
  - Authentic of attributes, trust credentials: SAML assertion to carry trust credentials
- **Distributed policy evaluation**: using Push model in AAA
- **Trust context description**:
  - Attribute profiles: using resource description languages
  - Semantics inference between attribute namespace ontologies
- **Dynamic trust relationships**
  - On-demand cloud resources
  - Provision trust policies

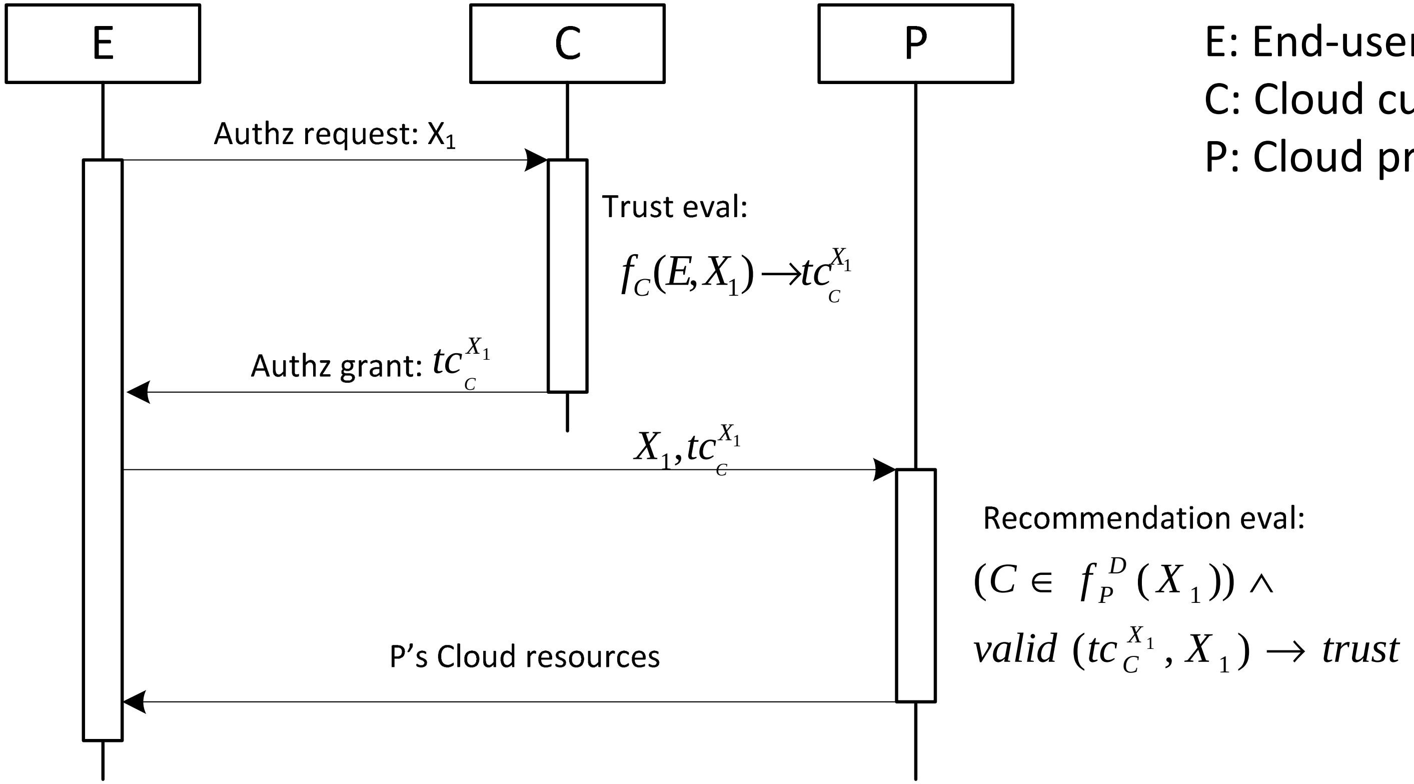
Dynamic Trust Establishment for Intercloud

- Use-case:
  - Consuming cloud resources from sub-contractor Cloud Service Providers
- Adopt cloud resources/services lifecycles
  - Request – Reservation – Deployment – Operation - Decommissioning
  - Reservation & Deployment phases
    - Establish direct trust relations between entities and/by linking/chaining trust anchors
    - Generate trust policies & delegation policies for provisioned cloud resources
    - Local attribute name spaces resolution
  - **Operation phase**
    - Establish (indirectdynamic) trust relationships for instantly provisioned infrastructures using trust policies & delegation policies

Indirect/Dynamic Trust Establishment Protocol

Operation phase:

Establish indirect trust relationships using trust policies & delegation policies


E
C
P
Authz request: X_1
Trust eval:
$f_C(E,X_1) \rightarrow tc_C^{X_1}$
Authz grant: $tc_C^{X_1}$
$X_1, tc_C^{X_1}$
Recommendation eval:
$(C \in f_P^D(X_1)) \wedge$
$valid\ (tc_C^{X_1}, X_1) \rightarrow trust$
P's Cloud resources

E: End-user

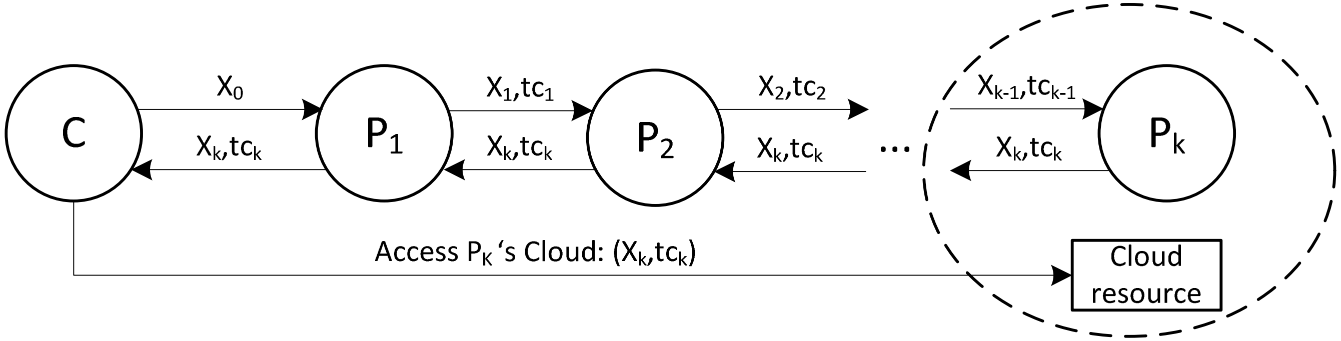
C: Cloud customer

P: Cloud provider

Indirect Trust Establishment Protocol Flow

Operation phase:

Establish indirect trust relationships for delegation chain of K providers (trust-chain)



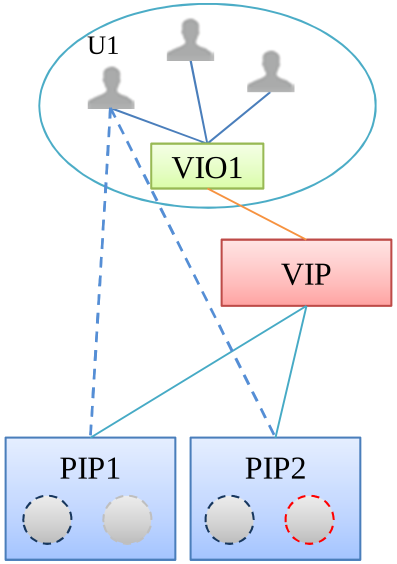
Indirect Trust Establishment Protocol Flow with Push Model

C: client

P_i: Cloud Providers i

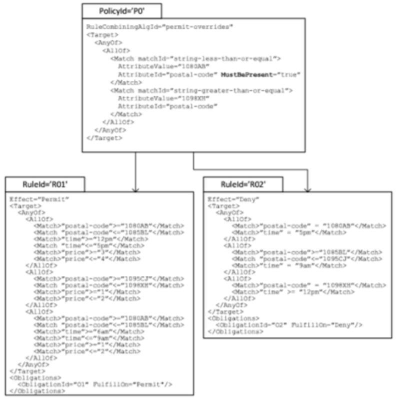
Implementation

- **Dynamic trust establishment protocol**: experiment in Geysers (https://geysers.eu)



◌ VM/storage at PIP

- **Trust evaluation engine**: SNEXACML
  - XACML extensions:
    - Policy issuer
    - Issuing trust credential: obligations
  - SAML assertion extension
  - Evaluation performance
    - Using Multi-type Interval Decision Diagrams (MIDD)



PolicyId='P0'

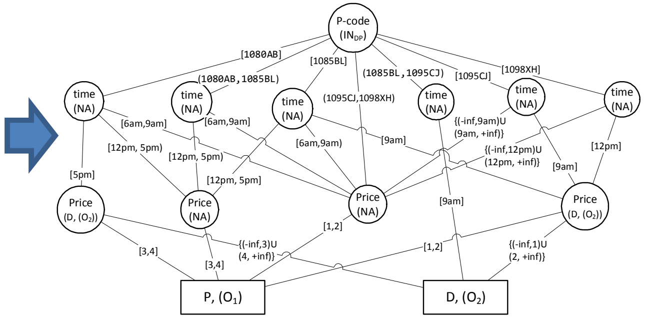
```
RuleCombiningAlgId="permit-overrides"
<Target>
  <AnyOf>
    <AllOf>
      <Match matchId="string-less-than-or-equal">
        AttributeValue="1080AB"
        AttributeId="postal-code" MustBePresent="true"
      </Match>
      <Match matchId="string-greater-than-or-equal">
        AttributeValue="1098XH"
        AttributeId="postal-code"
      </Match>
    </AllOf>
  </AnyOf>
</Target>
```

RuleId='R01'

```
Effect="Permit"
<Target>
  <AnyOf>
    <AllOf>
      <Match>"postal-code">="1080AB"</Match>
      <Match "postal-code"<="1085BL"</Match>
      <Match>"time">="12pm"</Match>
      <Match "time"<="5pm"</Match>
      <Match>"price">="3"</Match>
      <Match>"price"<="4"</Match>
    </AllOf>
    <AllOf>
      <Match>"postal-code">="1095CJ"</Match>
      <Match "postal-code"<="1098XH"</Match>
      <Match>"price">="1"</Match>
      <Match>"price"<="2"</Match>
    </AllOf>
    <AllOf>
      <Match>"postal-code">="1080AB"</Match>
      <Match "postal-code"<="1085BL"</Match>
      <Match>"time">="6am"</Match>
      <Match>"time"<="9am"</Match>
      <Match>"price">="1"</Match>
      <Match>"price"<="2"</Match>
    </AllOf>
  </AnyOf>
</Target>
<Obligations>
  <ObligationId="O1" FulfillOn="Permit"/>
</Obligations>
```

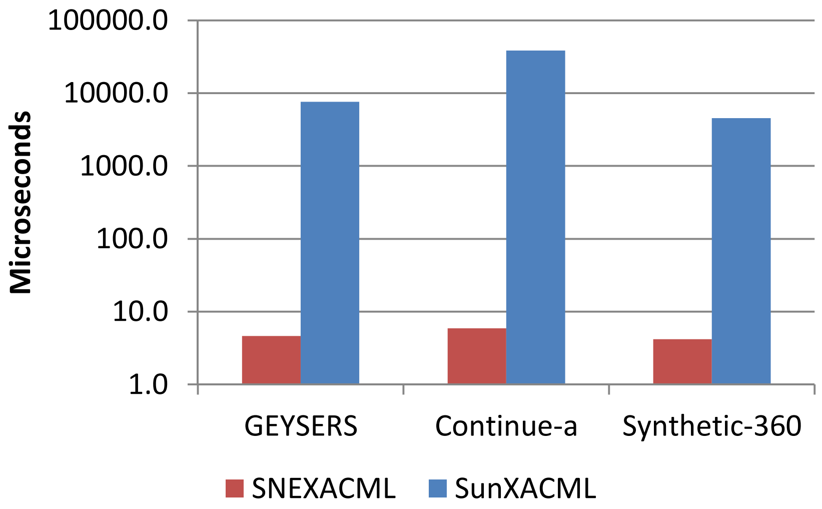
RuleId='R02'

```
Effect="Deny"
<Target>
  <AnyOf>
    <AllOf>
      <Match>"postal-code" = "1080AB"</Match>
      <Match>"time" = "5pm"</Match>
    </AllOf>
    <AllOf>
      <Match>"postal-code">="1085BL"</Match>
      <Match "postal-code"<="1095CJ"</Match>
      <Match>"time" = "9am"</Match>
    </AllOf>
    <AllOf>
      <Match>"postal-code" = "1098XH"</Match>
      <Match>"time" >= "12pm"</Match>
    </AllOf>
  </AnyOf>
</Target>
<Obligations>
  <ObligationId="O2" FulfillOn="Deny"/>
</Obligations>
```

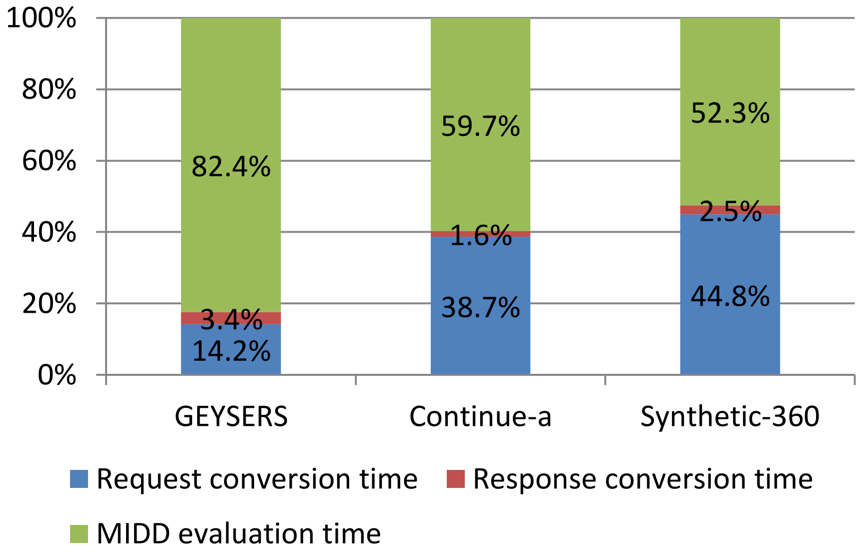


Trust evaluation engine: performance analysis

Datasets	Policy level	# Policy-sets	#Policies	#Rules	Attr	Operators
GEYSERS	3	6	7	33	3	=
Continue-a	6	111	266	298	14	=
Synthetic-360	4	31	72	360	10	=(80%), co-mplex(20%)



Average request evaluation time



Micro-benchmark evaluation response times

Conclusion

- An attribute-based approach for dynamic trust establishments for multiple Cloud providers
  - Attribute trust policies: flexible, manageable
  - Open for attribute namespaces resolutions
  - Dynamic provisioning trust relationships
  - High performance evaluation

Discussion and Future work

- On-going work
  - Resolutions of attribute namespaces ontologies
  - Attribute validation
  - Apply dynamic trust establishment protocol to Intercloud
  - Trust Policy Engine
- P2302 Group
  - Section 6.6-6.8, Intercloud Security
    - Trust Management Framework
      - Trust topology, protocols, evaluation mechanisms.
      - Auxiliary functions: collect and validate trust values, attributes, trust credentials

Thank you!

Contact Information

Canh Ngo, Yuri Demchenko
{t.c.ngo, y.demchenko}@uva.nl
System and Network Engineering research group (SNE)
University of Amsterdam