



RESTENA

TERENA - TF-EMC2

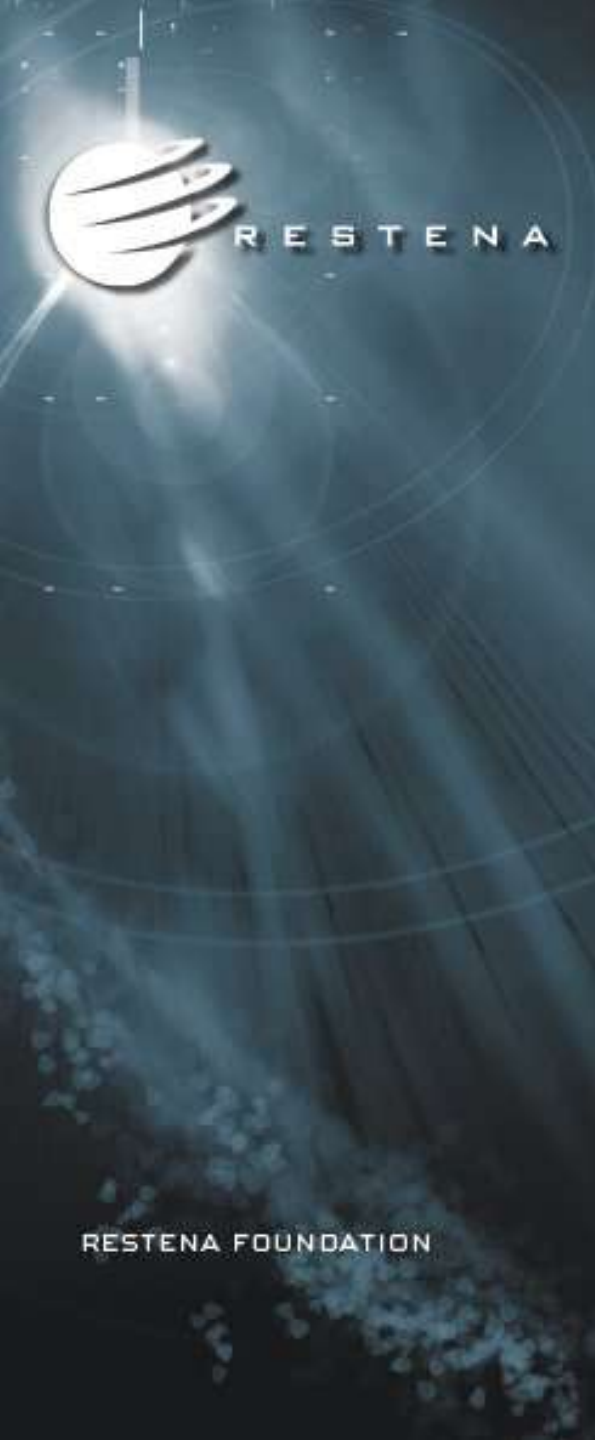
Use of SAML for authentication and
authorization in eduGAIN

Stefan Winter <stefan.winter@restena.lu>

Barcelona

8 September 2005

RESTENA FOUNDATION



SAML – Requests and Responses

- Completely XML-based (collection of XML Schema)
- Typical client-server architecture: one party asks (issues a request), the other party answers (issues a response)
- no unsolicited server-side messages
- Done via protocol elements <Request> and <Response>
- Covers Authentication, Attribute Release and Authorization

Fitting eduGAIN messages into SAML

- Abstract message parts are defined in the AAI architecture deliverable
- These are mapped into the existing SAML elements where possible; otherwise, SAML schema extensions are used
- Those abstract operations that are common for all interactions are mapped into Request/Response, the others into more specific Queries/Statements (later)



Common message parts

- Requests:
 - RequestID
 - Resource
- Responses:
 - ResponseID
 - InResponseTo
 - Resource
 - Result

SAML 1.1 <Request>

Color code for all slides:

red marks required attributes
blue marks optional attributes or elements
black marks mandatory elements

<Request **RequestID MajorVersion MinorVersion IssueInstant**>

<RespondWith^{0..n}>

<Signature>

<Query>

- XOR -

<SubjectQuery>

- XOR -

<AuthenticationQuery>

- XOR -

<AttributeQuery>

- XOR -

<AuthorizationDecisionQuery>

- XOR -

<AssertionIDReference>

- XOR -

<AssertionArtifact>

Generic extension point for custom extensions (in particular useful for an own implementation of a “Home Location” service)

very useful (obviously)

SAML 1.1 <Request> - parts used so far in eduGAIN

<Request RequestID MajorVersion MinorVersion IssueInstant>

<RespondWith>^{0..n}

<Signature>

<Query>

- XOR -

<SubjectQuery>

- XOR -

<AuthenticationQuery>

- XOR -

<AttributeQuery>

- XOR -

<AuthorizationDecisionQuery>

- XOR -

<AssertionIDReference>

- XOR -

<AssertionArtifact>

Generic extension point for custom extensions (in particular useful for an own implementation of a "Home Location" service)

very useful (obviously)

SAML 1.1 <Response>

<Response **ResponseID MajorVersion MinorVersion IssueInstant InResponseTo Recipient**>

<Signature>

<Status>

<StatusMessage>

<StatusDetail>

<StatusCode>

<StatusCode>

<Assertion **MajorVersion MinorVersion AssertionID Issuer IssueInstant**>

<Conditions>

<Advice>

<Signature>

<Statement>

- XOR -

<SubjectStatement>

- XOR -

<AuthenticationStatement>

- XOR -

<AuthorizationStatement>

- XOR -

<AttributeStatement>

1..n

Again the extension point ...

... and the statements

SAML 1.1 <Response> - parts used so far in eduGAIN

<Response ResponseID MajorVersion MinorVersion IssueInstant InResponseTo Recipient>

<Signature>

<Status>

<StatusMessage>

<StatusDetail>

<StatusCode>

<StatusCode>

Yes, we use nested StatusCode
-> see next slide

<Assertion MajorVersion MinorVersion AssertionID Issuer IssueInstant>

<Conditions>

<Advice>

<Signature>

<Statement>

- XOR -

<SubjectStatement>

- XOR -

<AuthenticationStatement>

- XOR -

<AuthorizationStatement>

- XOR -

<AttributeStatement>

1..n

Again the extension point ...

... and the statements

EduGAIN use of StatusCode

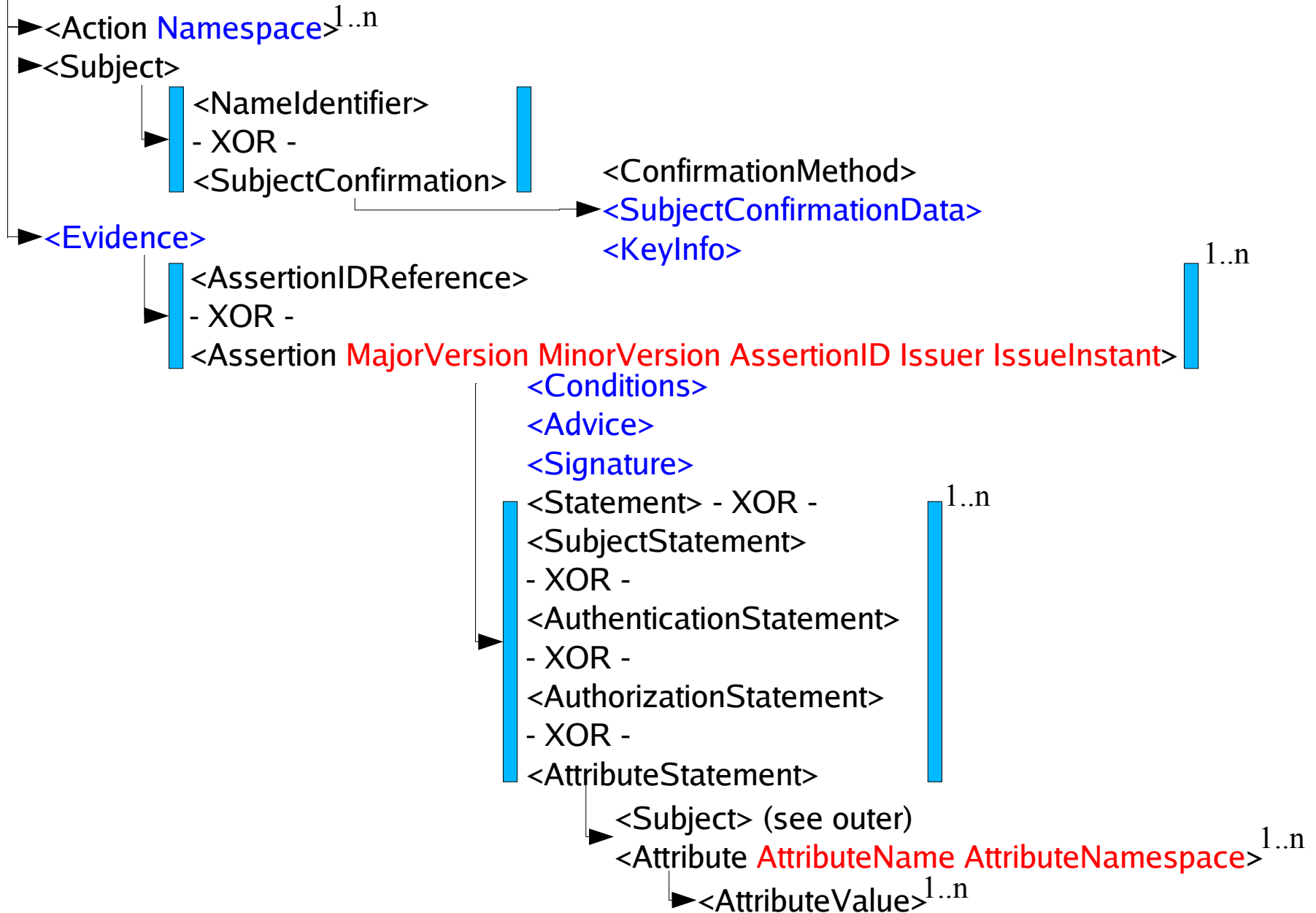
- Top-level <StatusCode> contains a string, but may only be filled with fixed values from the SAML protocol namespace
- eduGAIN needs to transport more codes than supported (extended error conditions)
- Nested <StatusCode> strings are free of restrictions
- So: use a generic (SAML-compliant) outer code and a more sophisticated inner code

Authorization Decisions - Request -

- Abstract operations in Authorization Request:
- General ones as seen before, plus:
 - Action
 - AttributeValueList
 - PolicyReference
 - CacheReference
 - Recipient
- Conveyed in the <AuthorizationDecisionQuery> element

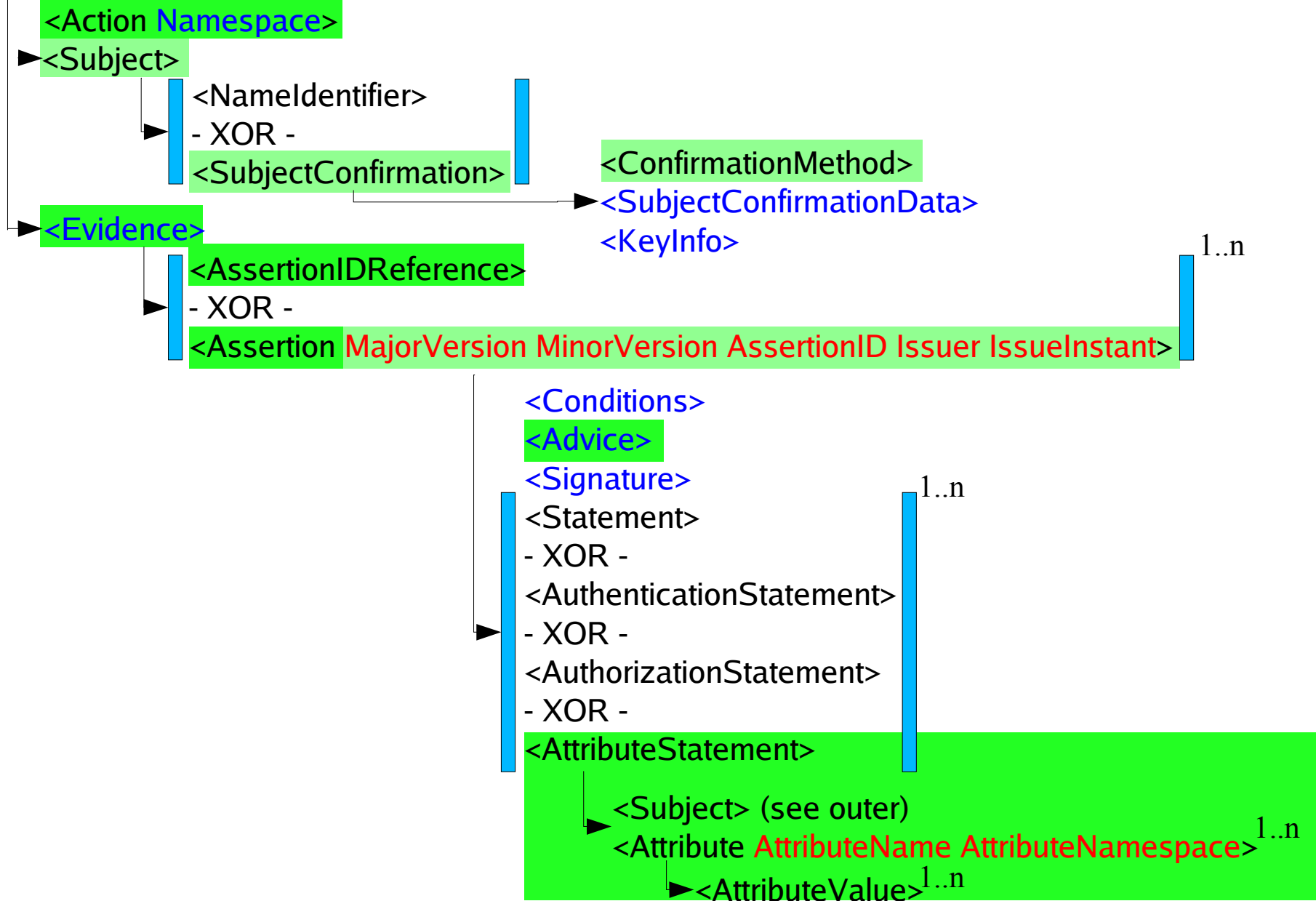
SAML 1.1 <AuthorizationDecisionQuery>

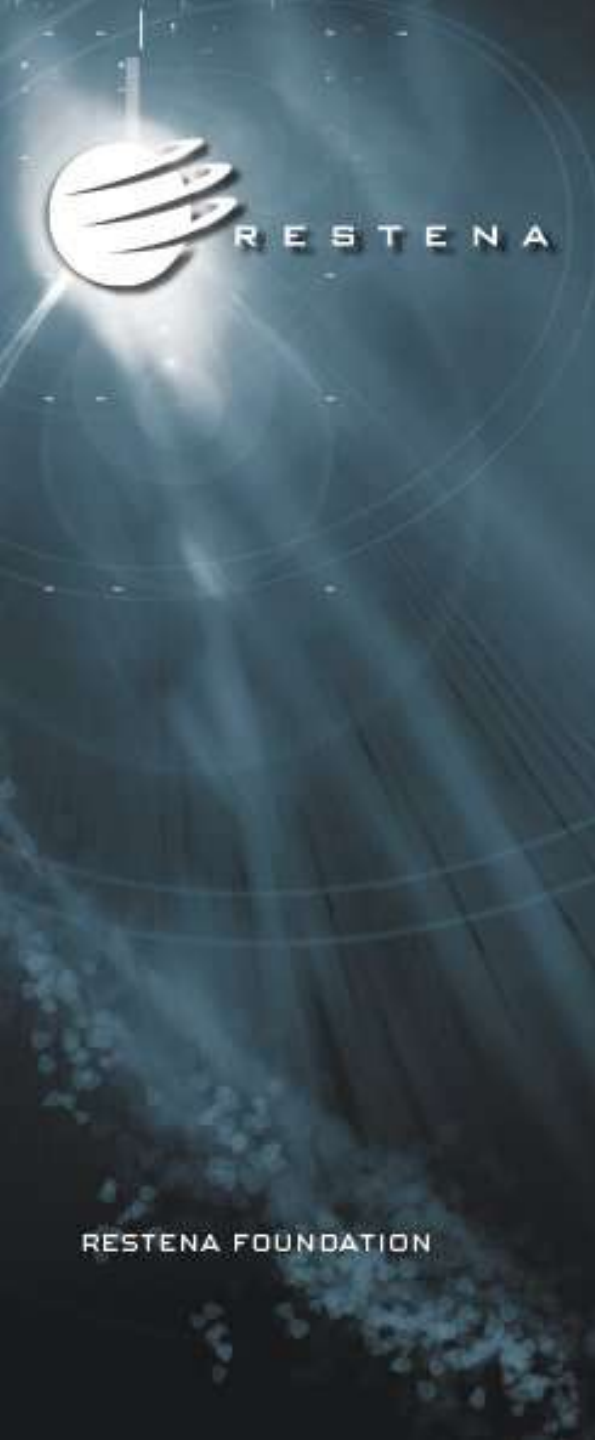
<AuthorizationDecisionQuery Resource>



SAML 1.1 <AuthorizationDecisionQuery> - eduGAIN parts

<AuthorizationDecisionQuery Resource>





AuthZ Query – missing part

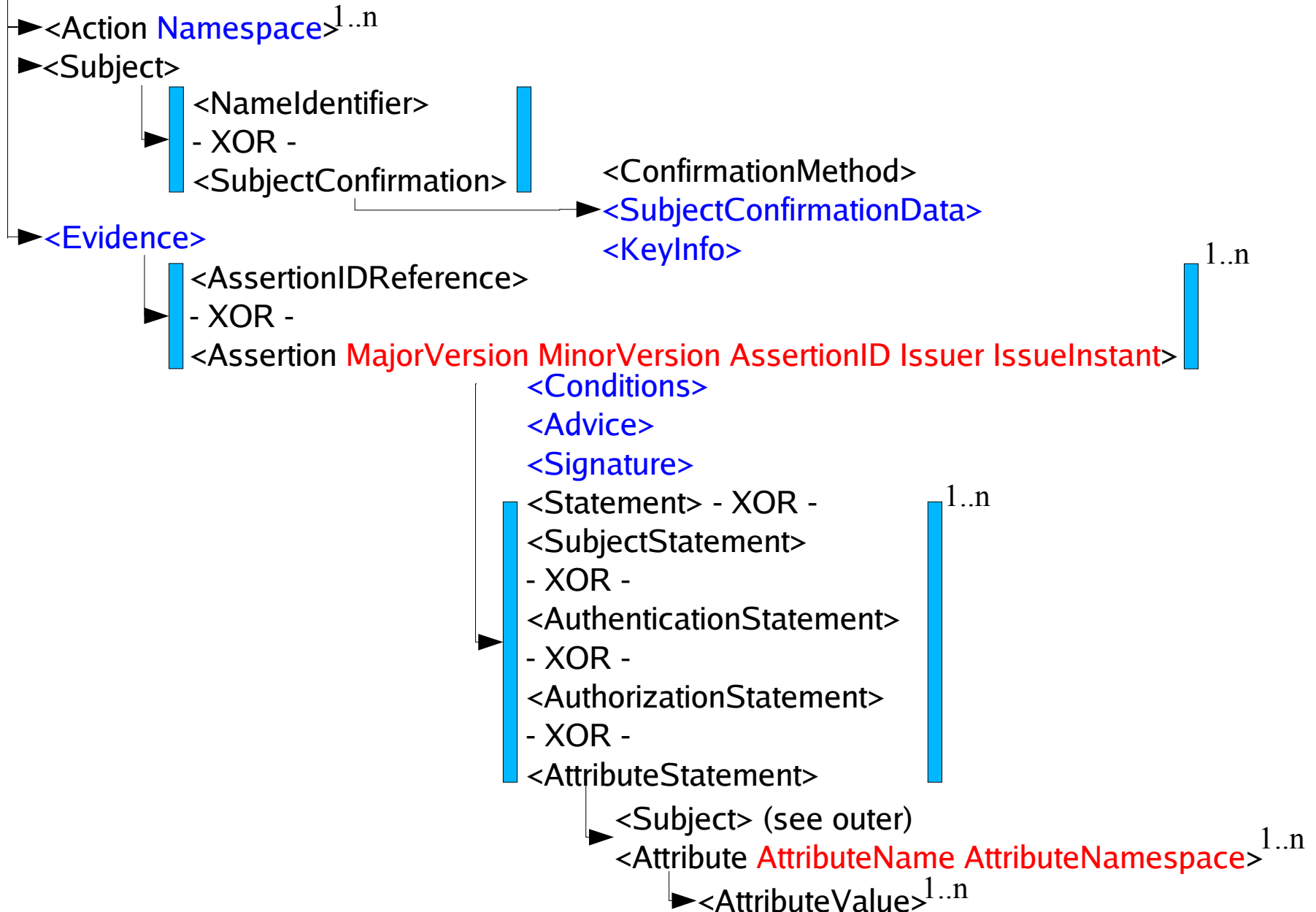
- Optional part “Recipient” has no place in the SAML request
- Solution: use XML Schema extension mechanism to allow this:
- `<ExtendedAuthorizationDecisionQuery>`
- derivate of standard query, allows `<Recipient>` as sub-element

Authorization Decisions - Response -

- Abstract operations in Authorization Response:
 - only the authorization decision
- Conveyed in
 - coarse status in the outer <Response> element (“Result”)
 - detailed list of authorized actions in <AuthorizationDecisionStatement> element

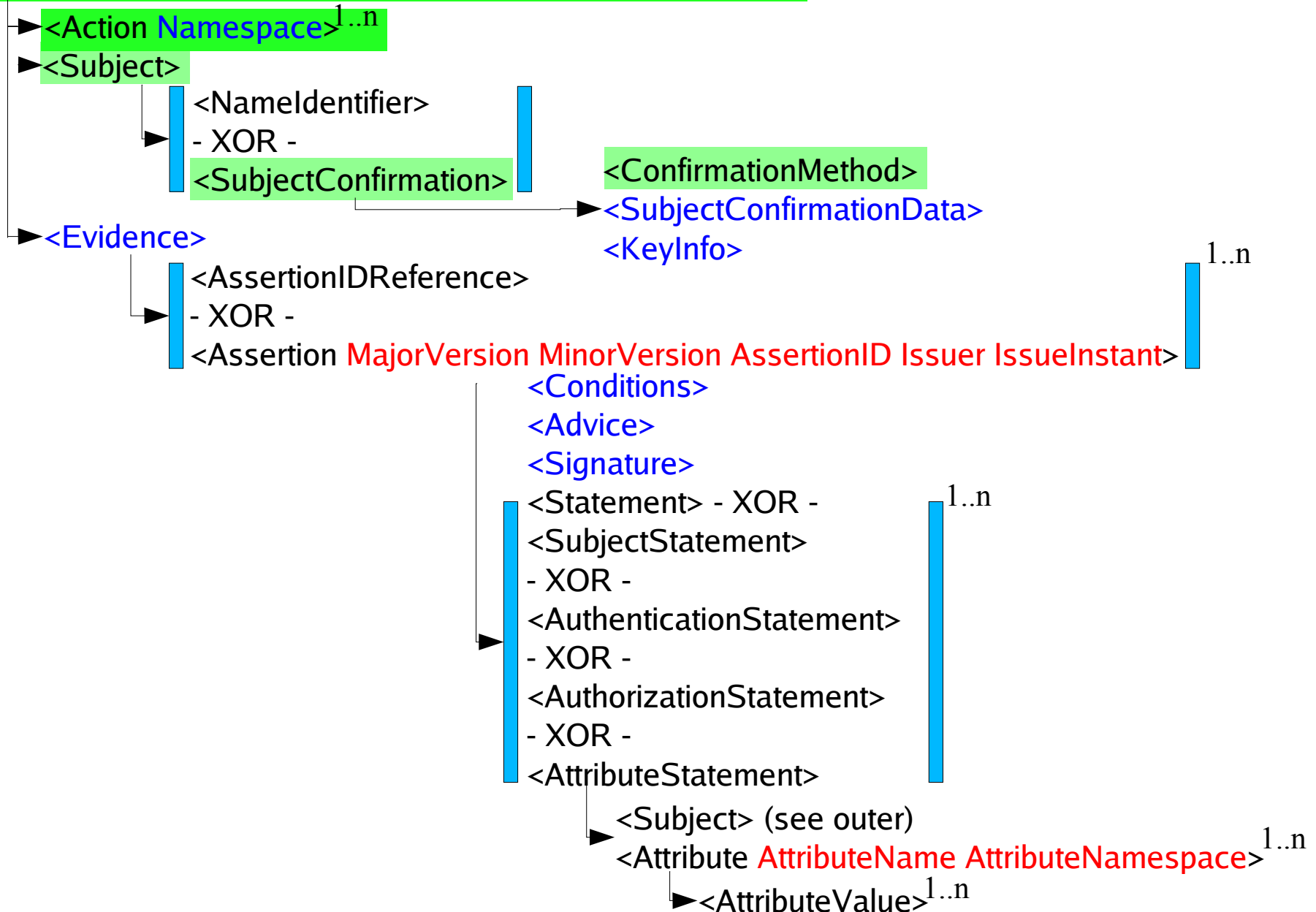
SAML 1.1 <AuthorizationDecisionStatement>

<AuthorizationDecisionStatement **Resource Decision**>



SAML 1.1 <AuthorizationDecisionStatement> - eduGAIN parts

<AuthorizationDecisionStatement Resource Decision>





RESTENA