KeyControls
1100100100001111111011010

Radboud Universiteit Nijmegen
Digital Security group

Using Idemix (IRMA card) in a federative scheme

Eric Verheul

Eric.Verheul@keycontrols.nl

Eric.Verheul@cs.ru.nl

IRMA meeting May 22, 2015

Agenda

- IRMA (Idemix) based ABC
- Basic federated authentication
- Federated IRMA I ("straightforward")
- Federated IRMA II: service provider convenient

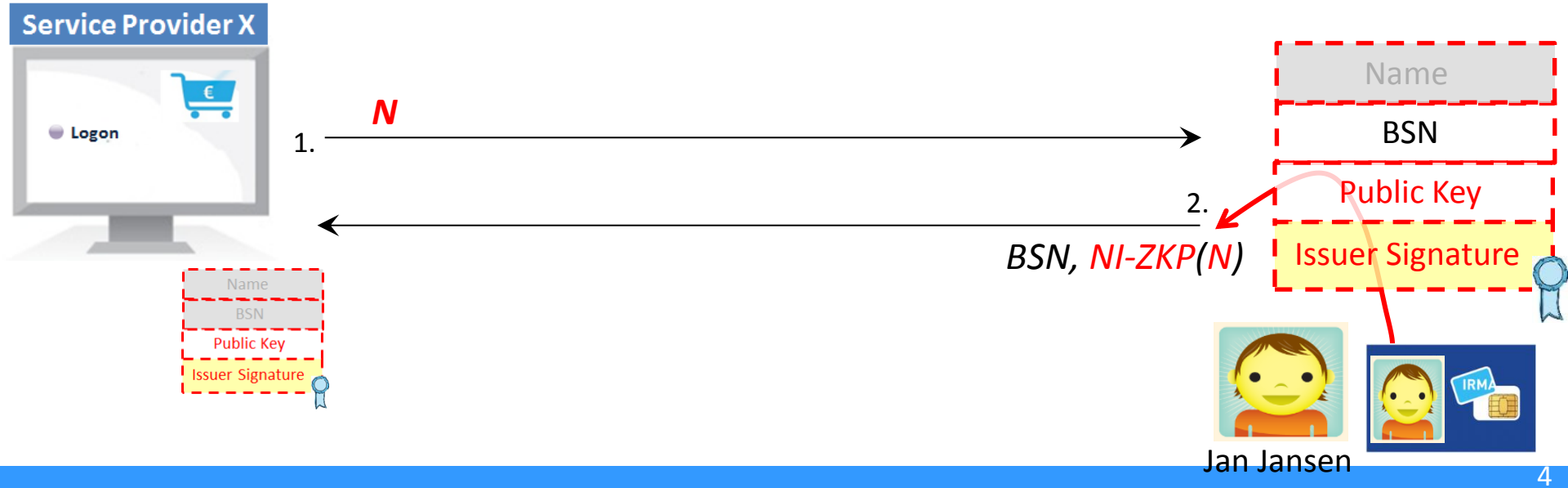
IRMA (Idemix) based ABC

- Direct electronic contact between IRMA card and Service Provider
- User sends IRMA certificate to Service Provider that contains encrypted attributes and a public key
- Certificate is first 'randomized' to prevent linkability issues
- User can reveal some attributes (and some not) and prove possession

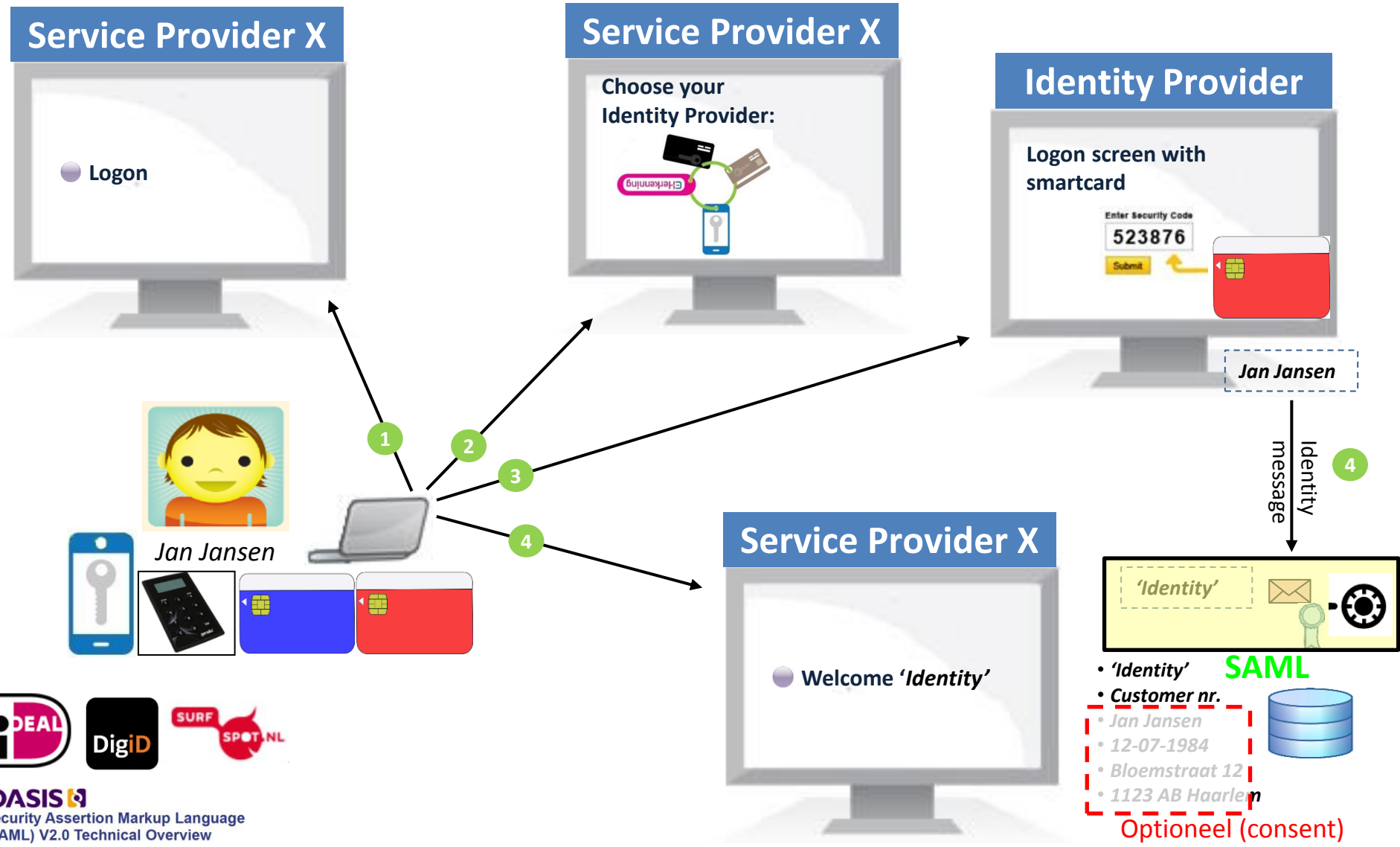
IRMA (Idemix) based ABC

- User reveals some attributes (e.g. BSN) and proves possession:
  1. Service Provider sends random number ***N*** to user
  2. User forms an attribute signature (NI-ZKP) proving:

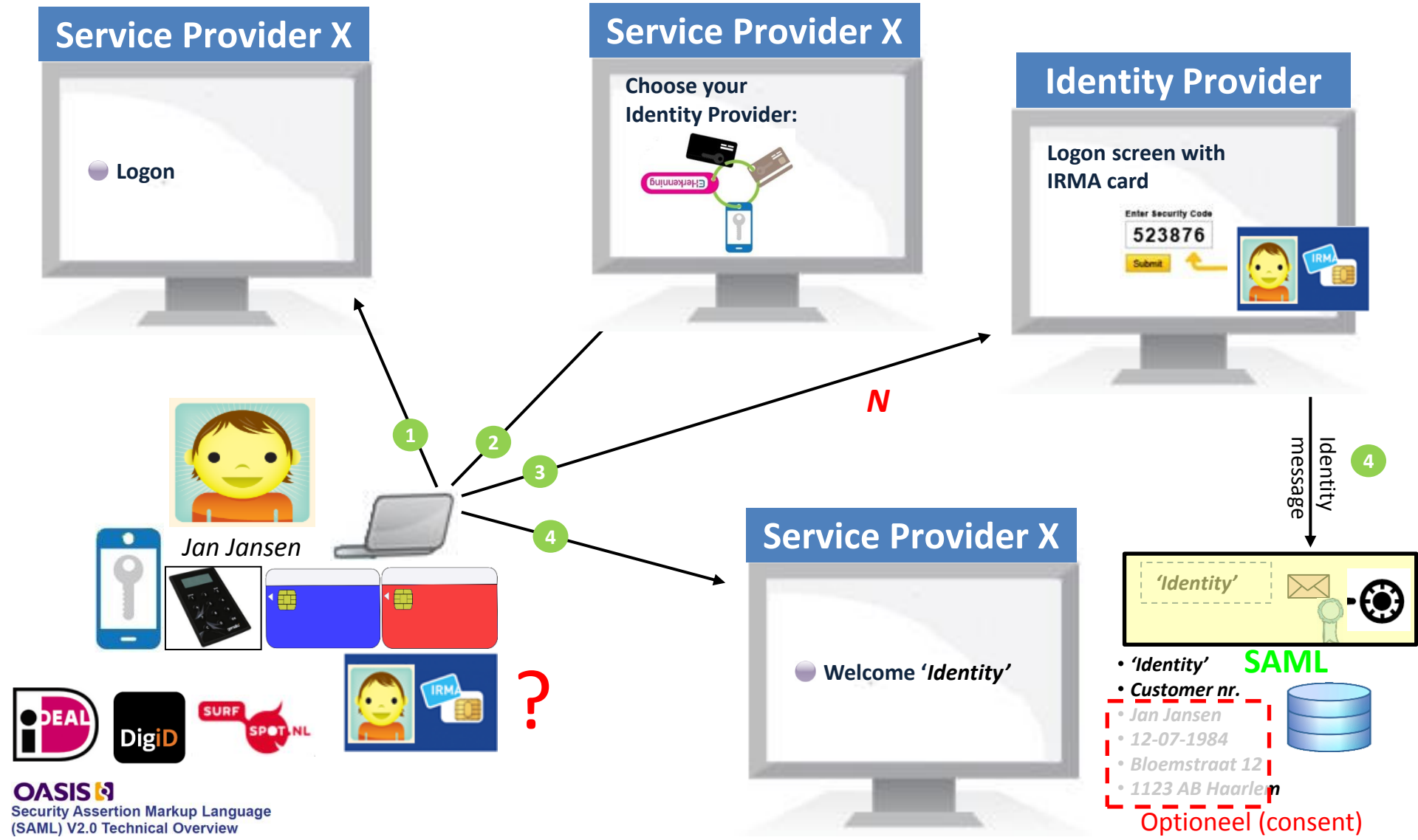
     "The attribute is inside the certificate and I own the certificate".
- Service Provider checks NI-ZKP and its 'freshness', i.e. based on ***N***.
- *Note: we ignore how the User would authenticate the SP*



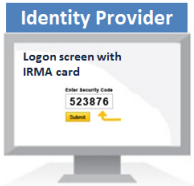
Basic federated authentication



Basic federated authentication



Federated IRMA I ("straightforward")

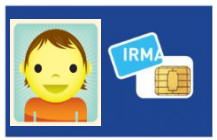


Name

BSN

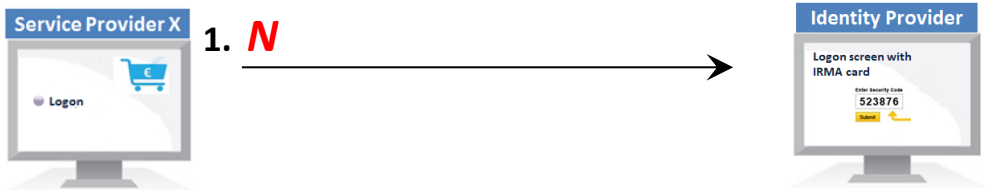
Public Key

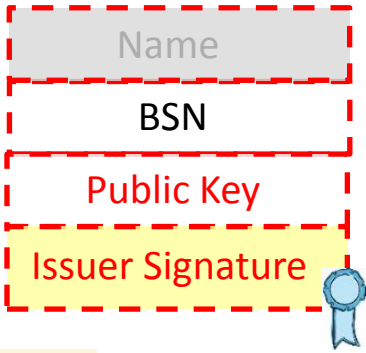
Issuer Signature

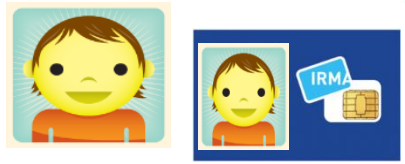


Federated IRMA I ("straightforward")

1. User is redirected to IdP with nonce ***N***

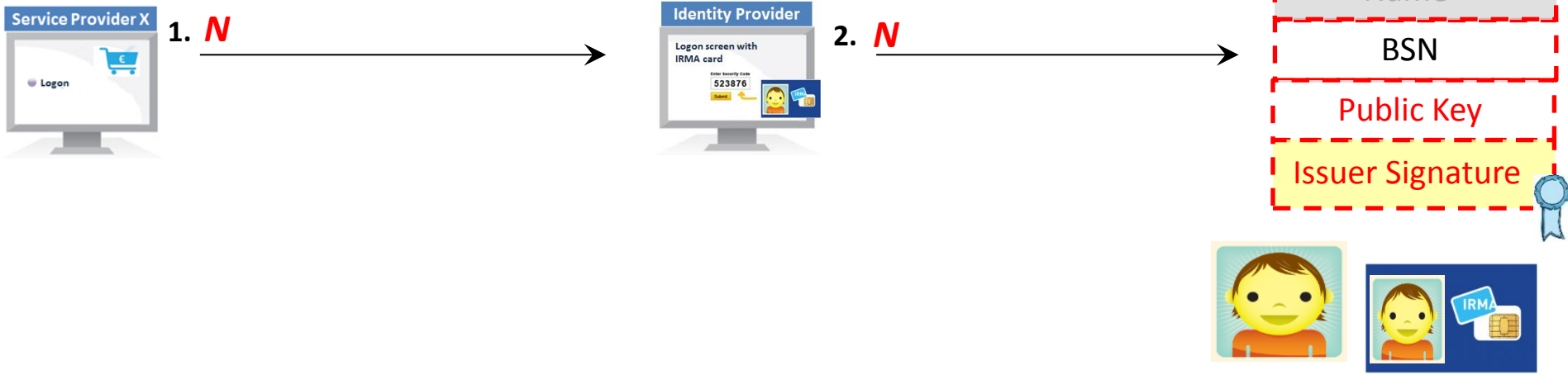






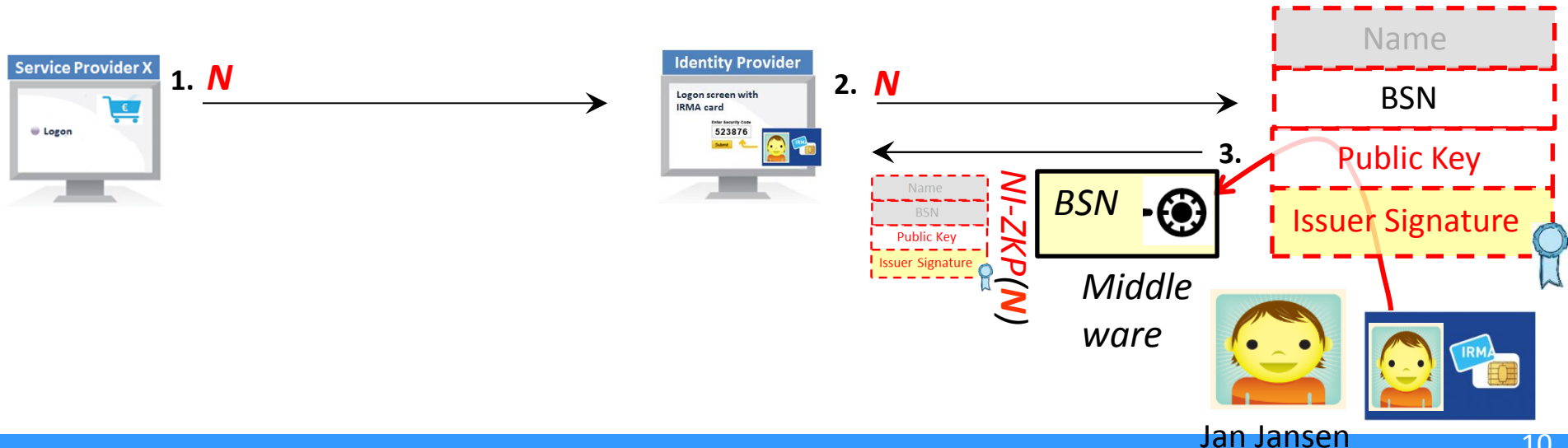
Federated IRMA I ("straightforward")

1. User is redirected to IdP with nonce ***N***
2. IdP sends ***N*** and asks User if he wants to reveal attribute (BSN) to SP X



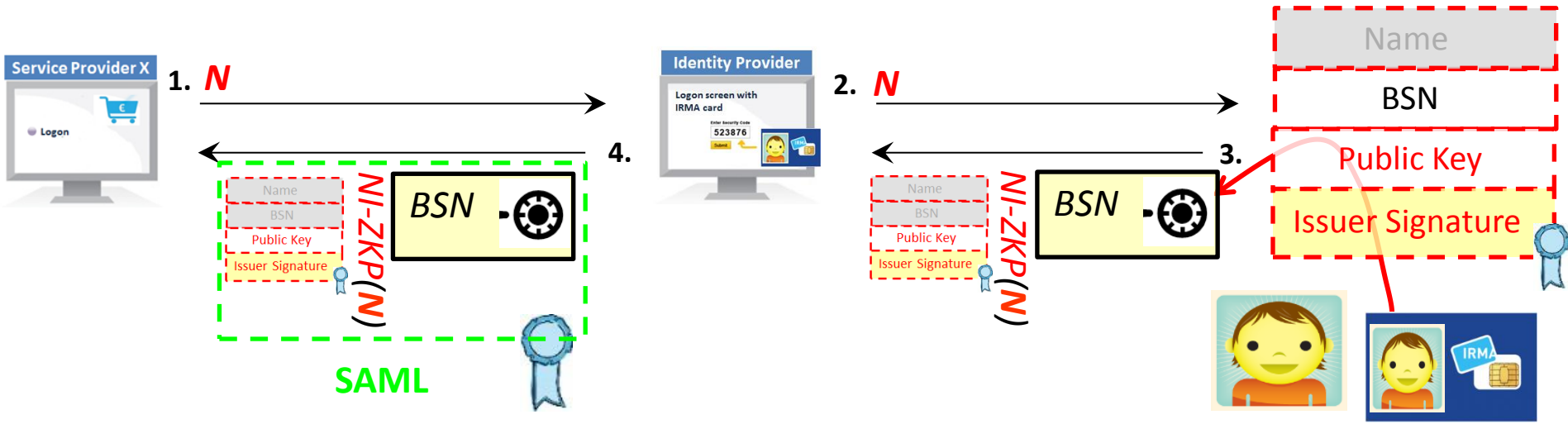
Federated IRMA I ("straightforward")

1. User is redirected to IdP with nonce ***N***
2. IdP sends ***N*** and asks User if he wants to reveal attribute (BSN) to SP X
3. If is this the case then User encrypts attribute with public key of SP X and sends this including randomized certificate and NI-ZKP(***N***) to IdP.



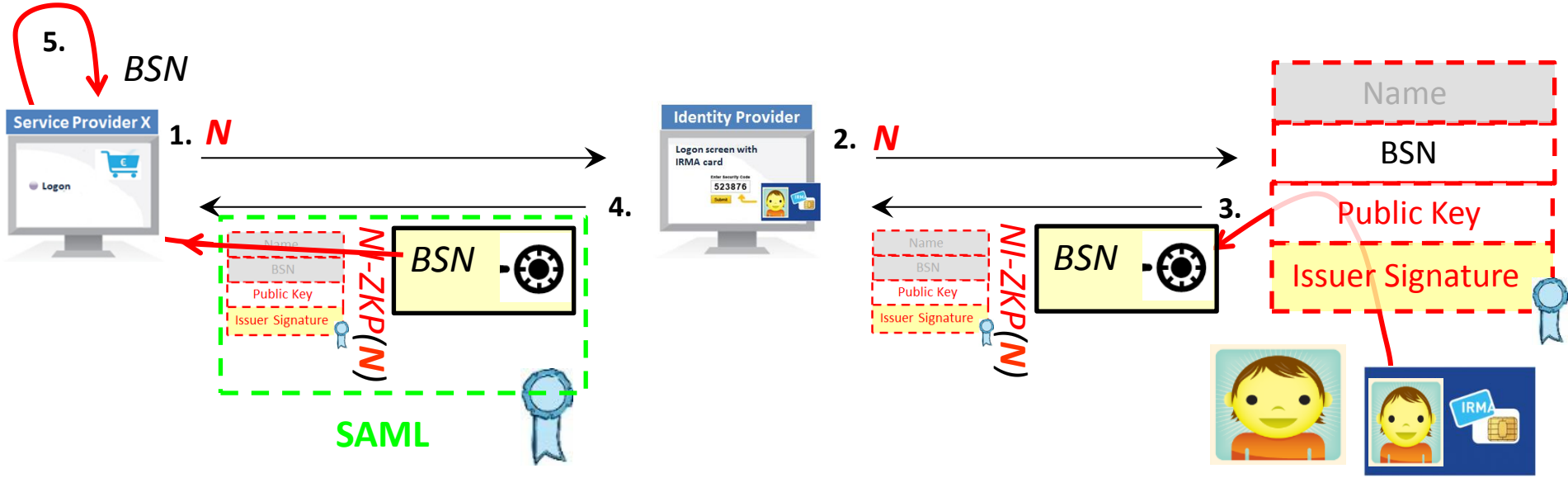
Federated IRMA I ("straightforward")

1. User is redirected to IdP with nonce ***N***
2. IdP sends ***N*** and asks User if he wants to reveal attribute (BSN) to SP X
3. If is this the case then User encrypts attribute with public key of SP X and sends this including randomized certificate and NI-ZKP(***N***) to IdP
4. IdP wraps encrypted message, NI-ZKP(***N***) and certificate in a signed SAML identity message and sends it to the SP.



Federated IRMA I ("straightforward")

1. User is redirected to IdP with nonce ***N***
2. IdP sends ***N*** and asks User if he wants to reveal attribute (BSN) to SP X
3. If is this the case then User encrypts attribute with public key of SP X and sends this including randomized certificate and NI-ZKP(***N***) to IdP
4. IdP wraps encrypted message, NI-ZKP(***N***) and certificate in a signed SAML identity message and sends it to the SP.
5. SP decrypts attribute and and checks it using NI-ZKP(***N***), including that it is 'fresh', i.e. based on ***N***.



Federated IRMA I ("straightforward")

1. By using standard asymmetric cryptography (probabilistic) the IdP would not be able to link the user through the encrypted BSN (and NI-ZKP) . The Identity Provider is just a proxy.
2. The Service Provider has possession of the attribute BSN but has ***no*** guarantee that it is authentic unless he checks the NI-ZKP. This means that Service Provider is relieved from connecting to the IRMA device, but would still need to run all Idemix checks which is cumbersome.
3. It would be more convenient if the Identity Provider could relieve (‘ontzorgen’) the Service Provider even further, i.e. making the check that the BSN is authentic as simple as possible.

Federated IRMA II: service provider convenient (regular use of IRMA card by SP in detail)

$S, Z, R, R_1, \ldots, R_L \in_R \mathrm{QR}_n$ public, fixed.

$U = S^v \cdot R^m$ user "public key".

$$Z \cdot R_1^{-\mathrm{BSN}} = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i}$$

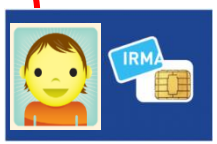
1. *N*

2. *BSN, NI-ZKP(N)*

1. 1.

2.

Name
BSN
Public Key
Issuer Signature



Jan Jansen

Federated IRMA II: service provider convenient

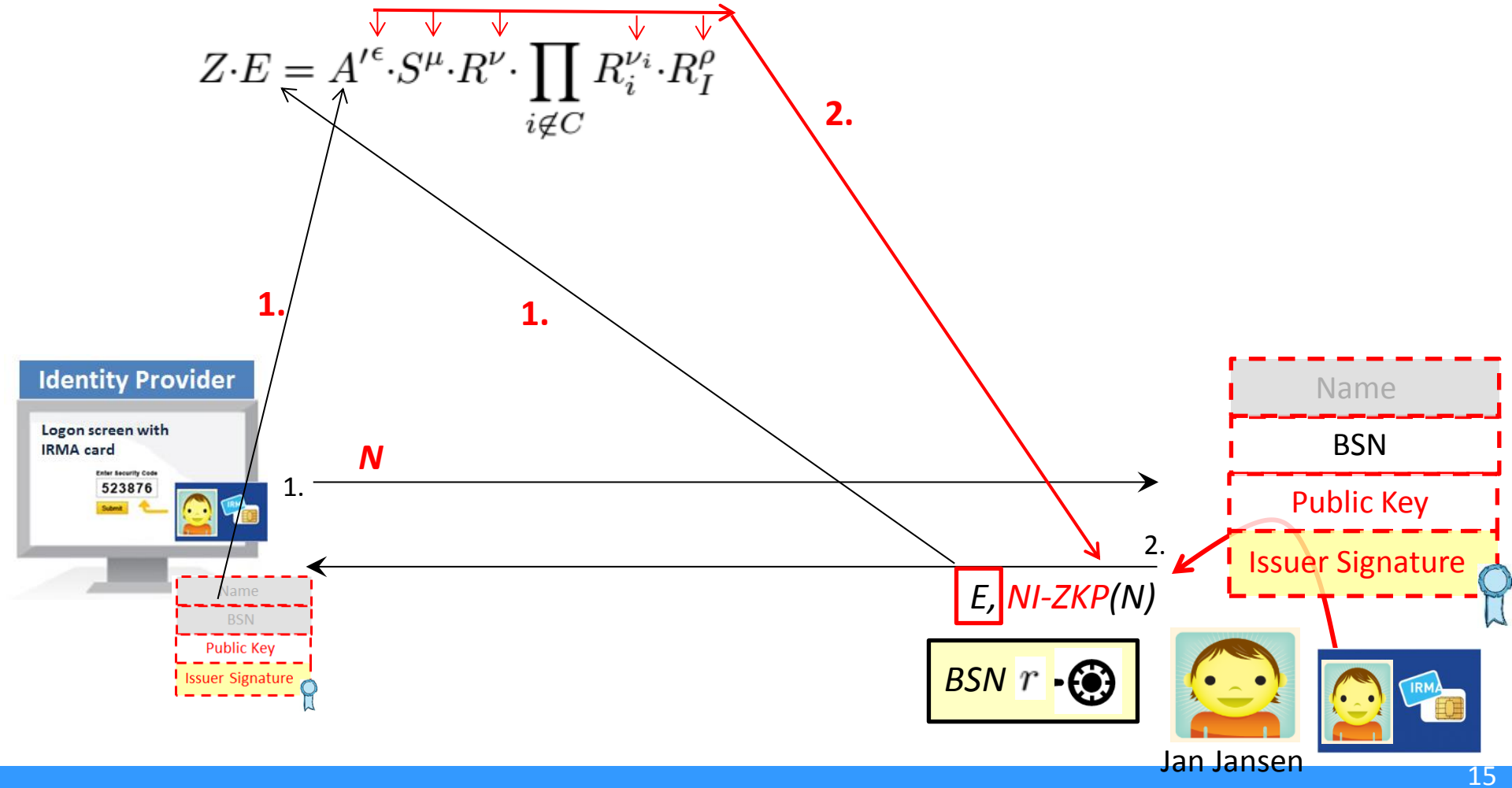
$S, Z, R, R_1, \ldots, R_L \in_R \mathrm{QR}_n$ public, fixed.

$U = S^v \cdot R^m$ user "public key".

Introduce extra $R_I \in_R \mathrm{QR}_n$.

Card produces $E = R_1^{-\mathrm{BSN}} \cdot R_I^r$ with r random (=encryption of BSN).

$$Z \cdot E = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i} \cdot R_I^{\rho}$$



Federated IRMA II: service provider convenient

$S, Z, R, R_1, \ldots, R_L \in_R \mathrm{QR}_n$ public, fixed.

$U = S^v \cdot R^m$ user "public key".

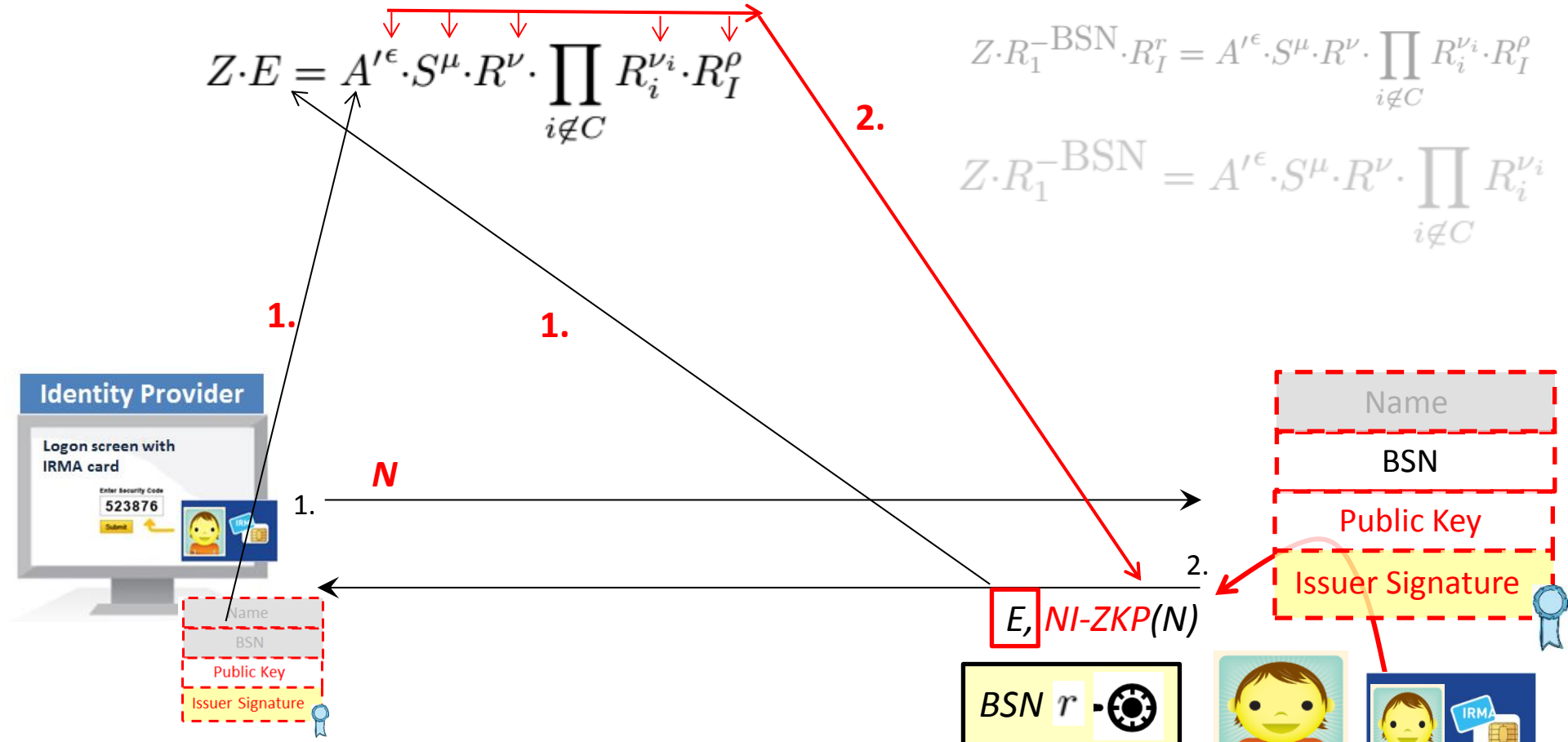
Introduce extra $R_I \in_R \mathrm{QR}_n$.

Card produces $E = R_1^{-\mathrm{BSN}} \cdot R_I^r$ with r random (=encryption of BSN).

$$Z \cdot E = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i} \cdot R_I^{\rho}$$

$$Z \cdot R_1^{-\mathrm{BSN}} \cdot R_I^r = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i} \cdot R_I^{\rho}$$

$$Z \cdot R_1^{-\mathrm{BSN}} = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i}$$



Federated IRMA II: service provider convenient

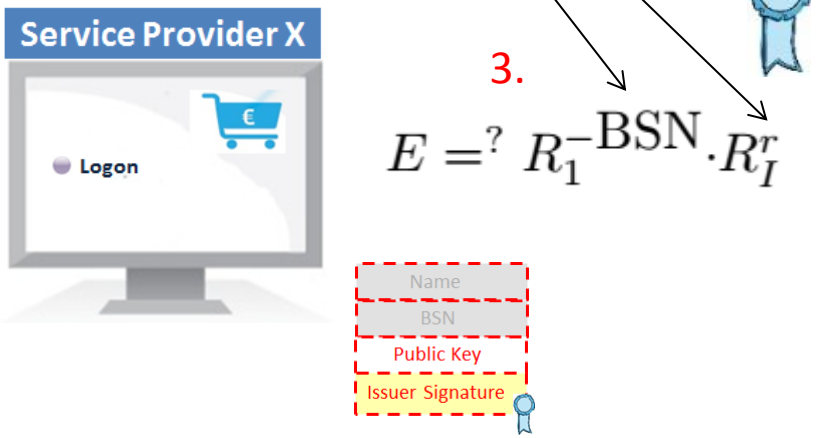
$S, Z, R, R_1, \ldots, R_L \in_R \mathrm{QR}_n$ public, fixed.

$U = S^v \cdot R^m$ user "public key".

Introduce extra $R_I \in_R \mathrm{QR}_n$.

Card produces $E = R_1^{-\mathrm{BSN}} \cdot R_I^r$ with r random (=encryption of BSN).

$$Z \cdot E = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i} \cdot R_I^{\rho}$$



Federated IRMA II: service provider convenient

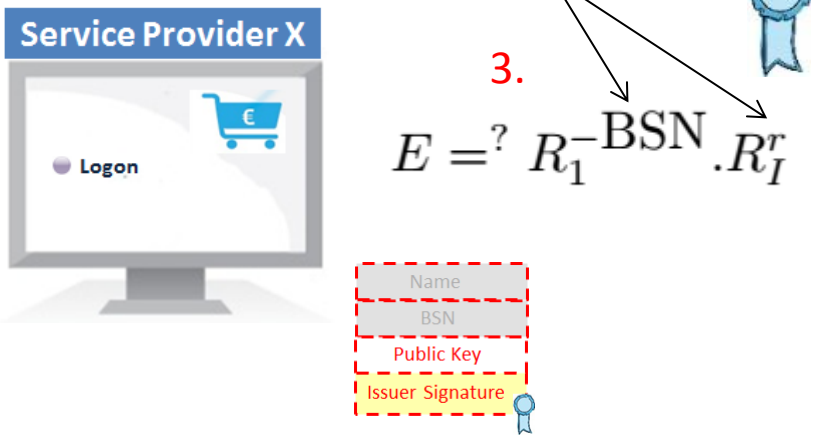
$S, Z, R, R_1, \ldots, R_L \in_R \mathrm{QR}_n$ public, fixed.

$U = S^v \cdot R^m$ user "public key".

Introduce extra $R_I \in_R \mathrm{QR}_n$.

Card produces $E = R_1^{-\mathrm{BSN}} \cdot R_I^r$ with r random (=encryption of BSN).

$$Z \cdot E = A'^{\epsilon} \cdot S^{\mu} \cdot R^{\nu} \cdot \prod_{i \notin C} R_i^{\nu_i} \cdot R_I^{\rho}$$



- If Service Provider trusts IdP check 3. suffices.
- Check 3. consists of two static RSA encryptions and one multiplication modulo the RSA modulus.
- Service Provider can also check on the IdP with NI-ZKP (or has proof IdP did a bad job).

Questions?