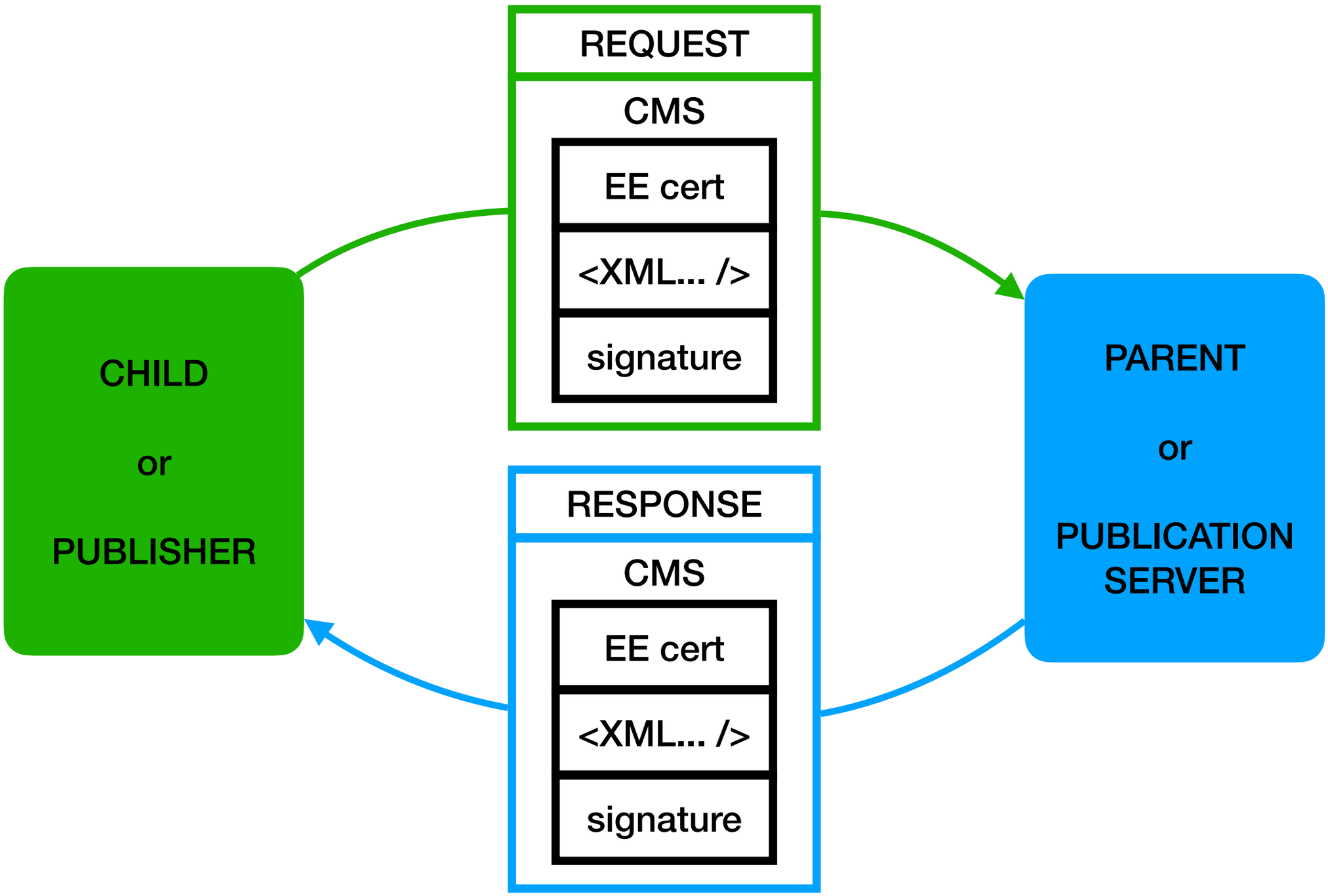
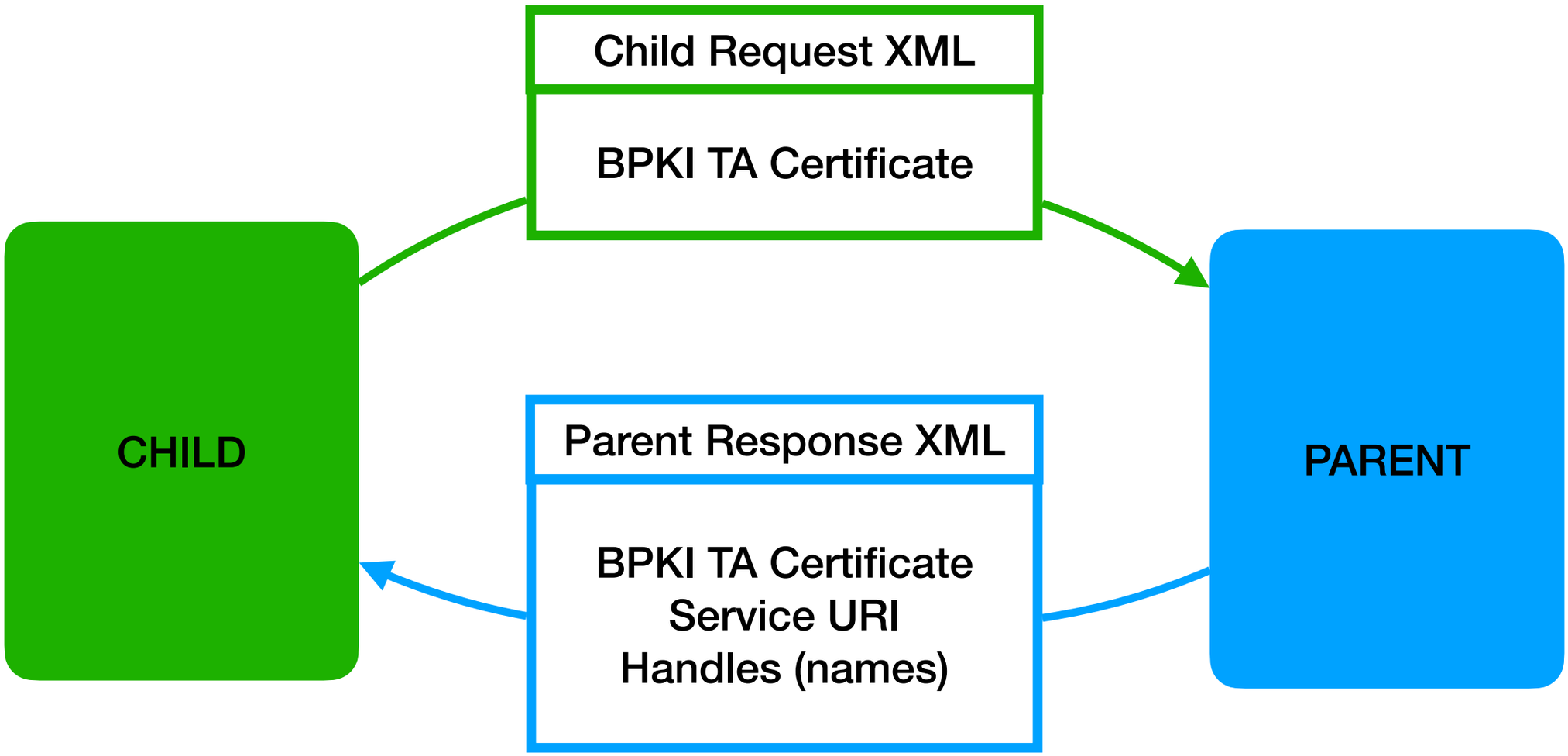
RFC 8183 BPKI Key Rollovers

Tim Bruijnzeels - IETF 116

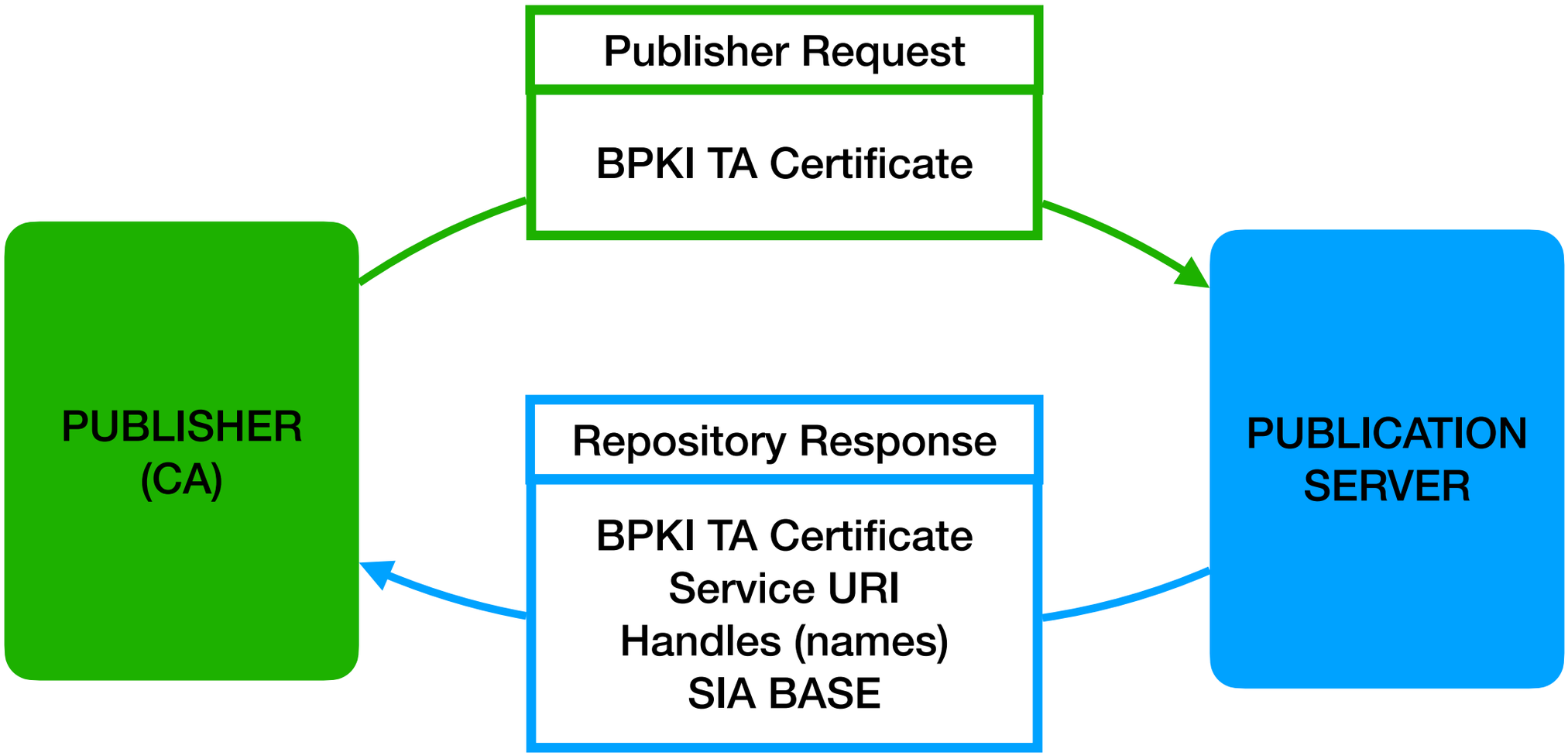
BPKI Keys?


REQUEST
CMS
EE cert
<XML... />
signature
CHILD
or
PUBLISHER
PARENT
or
PUBLICATION SERVER
RESPONSE
CMS
EE cert
<XML... />
signature

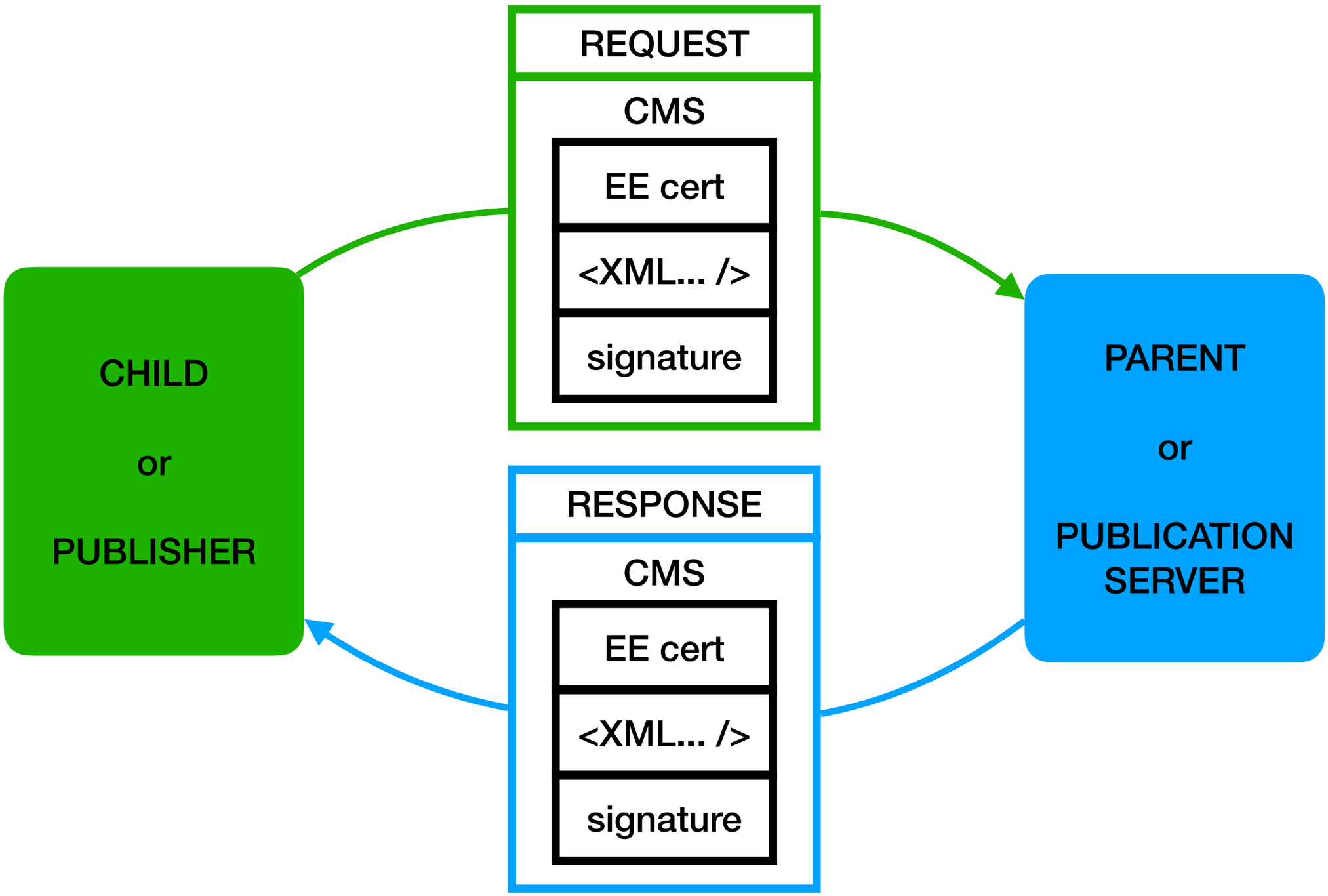
RFC 8183 Exchanges



RFC 8183 Exchanges



New BPKI Keys?



Easy, right?

Just exchange a couple of files...

Wrong..

Same key used for different relations

Planned/Unplanned

Child != Parent
Publisher != Server

When can the new key be used?

When can the old key be deleted?

How do different actors learn?

New service uri? New SIA base?!

...

Current Options Child/Publisher

Generate new key pair and request XML

Some CA and Publication Server implementations allow replacing just the self-signed BPKI TA certificate by uploading a new request or response XML. This is relatively painless, but there is a brief window where:

- ➡ **child cannot request certificate**
- ➡ **child cannot publish updates**

Other implementations do not support this:

- ➡ **Need to revoke and quickly add child / publisher**
- ➡ **CA certificate for child and objects disappear (briefly)**

Current Options Parent / Publication Server

Limited..

The child CA implementation may allow replacing just the self-signed BPKI TA certificate by uploading a new response XML.

Problems:

- ➡ **Timing with many children (they need to take action)**
- ➡ **Keep track of which key to use for each child**

Planned Roll Child

Minimise operator actions to a new exchange of RFC 8183 XML files.

GENERATE NEW KEY PAIR	Child creates Child Request XML with tag. Keep using old key for signing, for now.
NEW XML TO PARENT	Parent creates new Parent Response XML matching request tag. Accepts both keys.
NEW XML TO CHILD	After seeing response matching request tag: delete old key pair and use new key for signing.
DEPRECATE OLD KEY PAIR PARENT	Child is seen using new key.. no longer accept old key.

Planned Roll Parent

GENERATE NEW KEY PAIR	Parent generates new key pair and parent response with new identifier (modified service URI path or handle) for each child. Uses old key for old identifier, new key for new identifier.
ALERT CHILDREN	Email.. meetings.. phone.. media..
NEW XML TO CHILD	Child uses new response. Contacts parent on using new identifier (URI path / handle) and expects new parent key.
EACH CHILD	Parent tracks acceptance of new key, parent deletes old key when all children accepted.
PATIENCE..	Parent loses patience.. signs long lived CMS with error response(*) with old key and deletes key. Or... perhaps just uses new key.

***: Error message: "We told you 15 times we have a new key! See <url>"**

Planned Roll Publisher / Publication Server

Very similar to child and parent CA.

But note that the "sia_base", which determines where RPKI objects may be published, MUST NOT change. Doing so would be much more involved - essentially requiring a repository migration on the publisher part.

Automation?

What if you have 1500 child CAs or Publishers?

Similar to the normal process, but use in-band signalling. Add an "acceptance timer" (similar to signed TAL) to mitigate issues around temporary key access by an (on path) attacker.

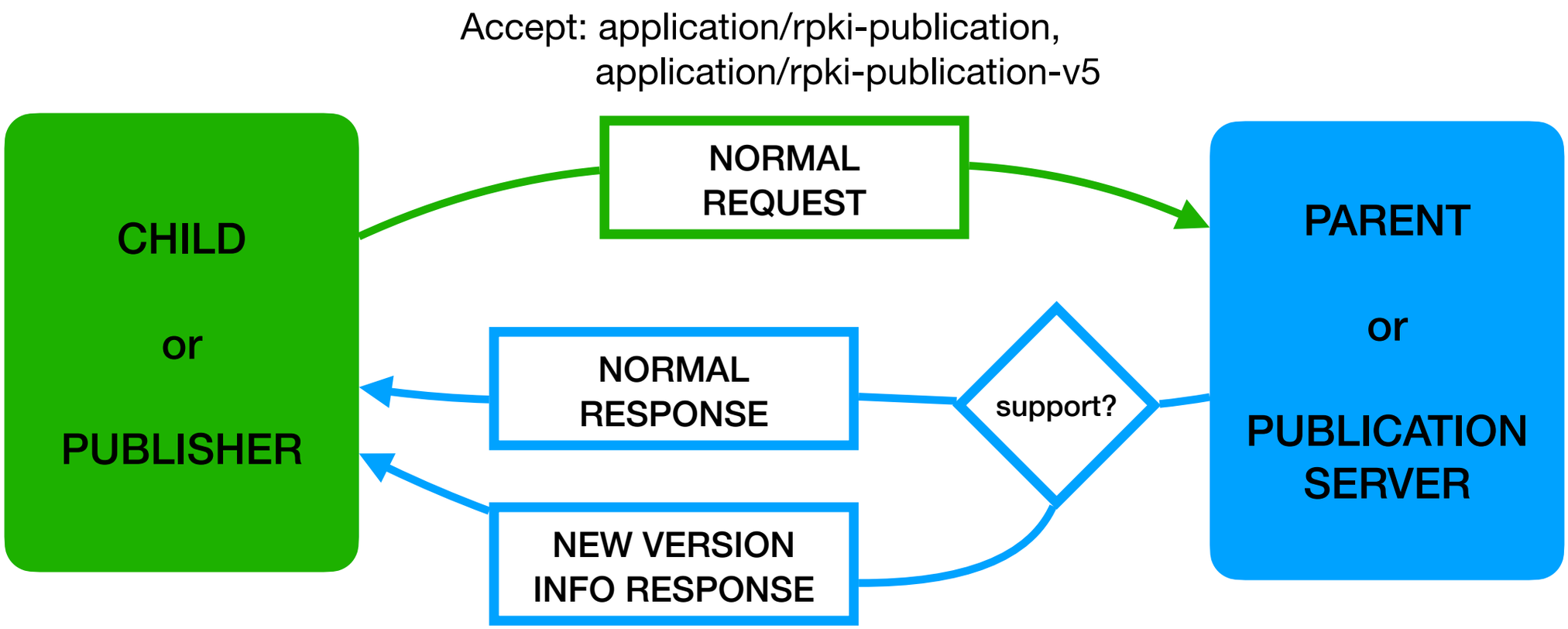
Only support planned key rollovers. Fall back to manual process in case of key compromises or any other issue.

Needs extensions to RFC 6492/8181. Leave no one behind:

- ➡ **Graceful protocol version negotiation**
- ➡ **Manual process still exists**

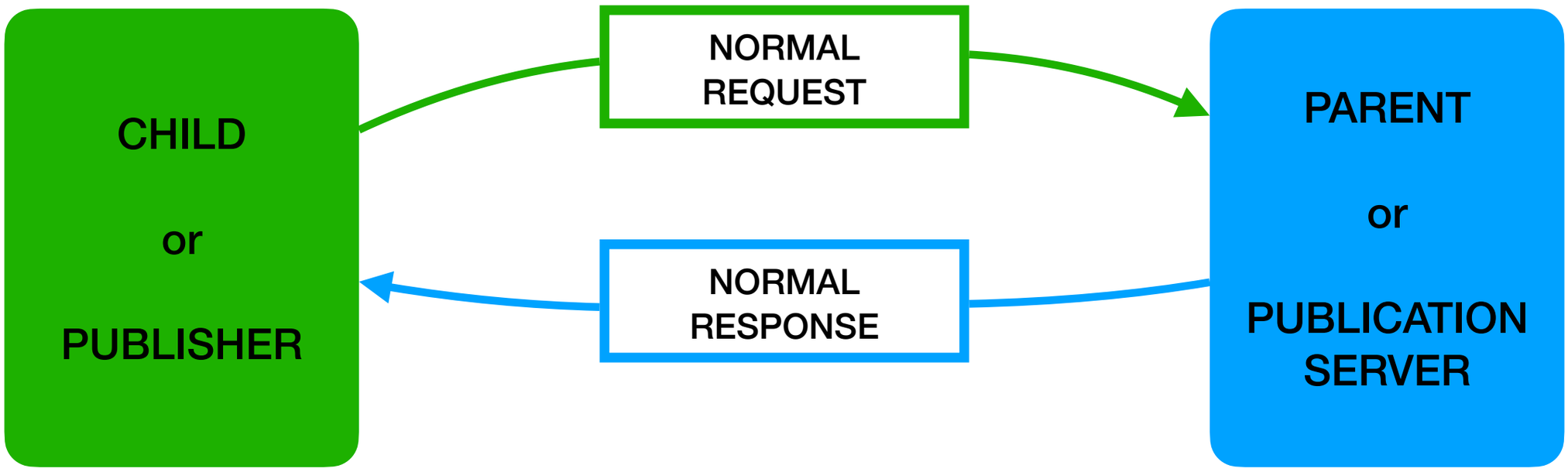
Version Negotiation

Option A: Child/Publisher probes for capability using headers



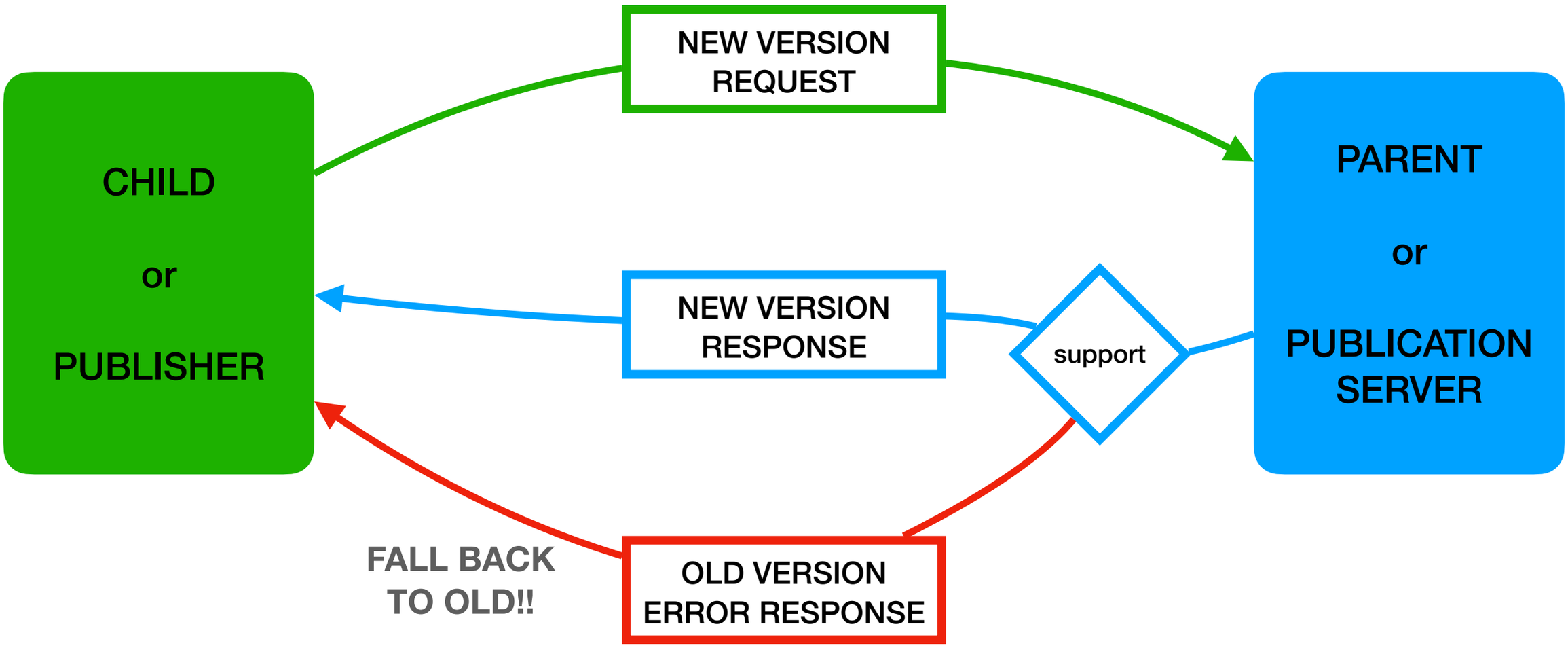
Version Negotiation

Option B: Use non-critical extension in response CMS EE

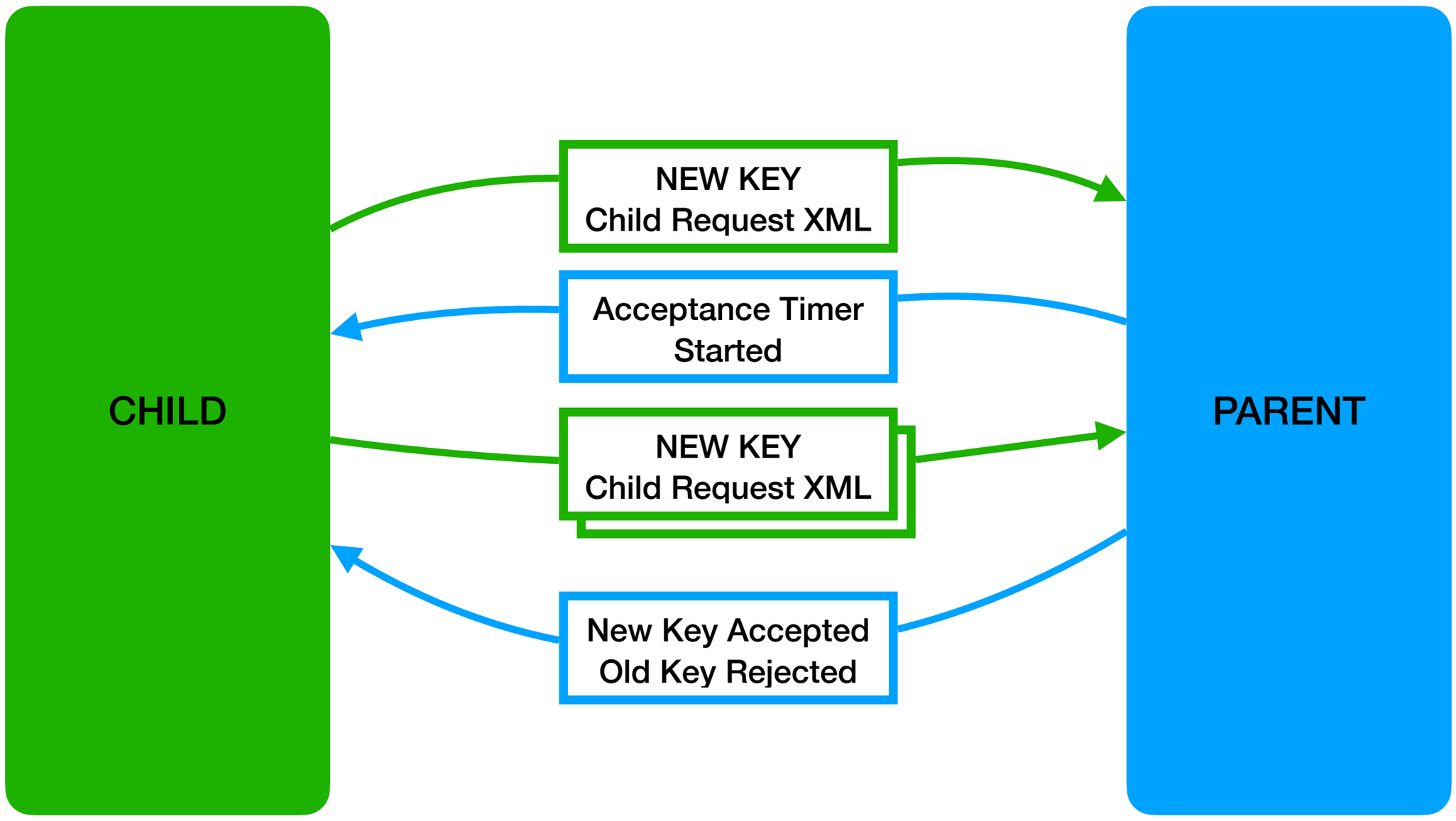


Child/Publisher inspects EE certificate in response CMS. If a "version-support" extension (TBD) is found, then upgrade to the highest possible version on the next exchange.

New Version Exchange

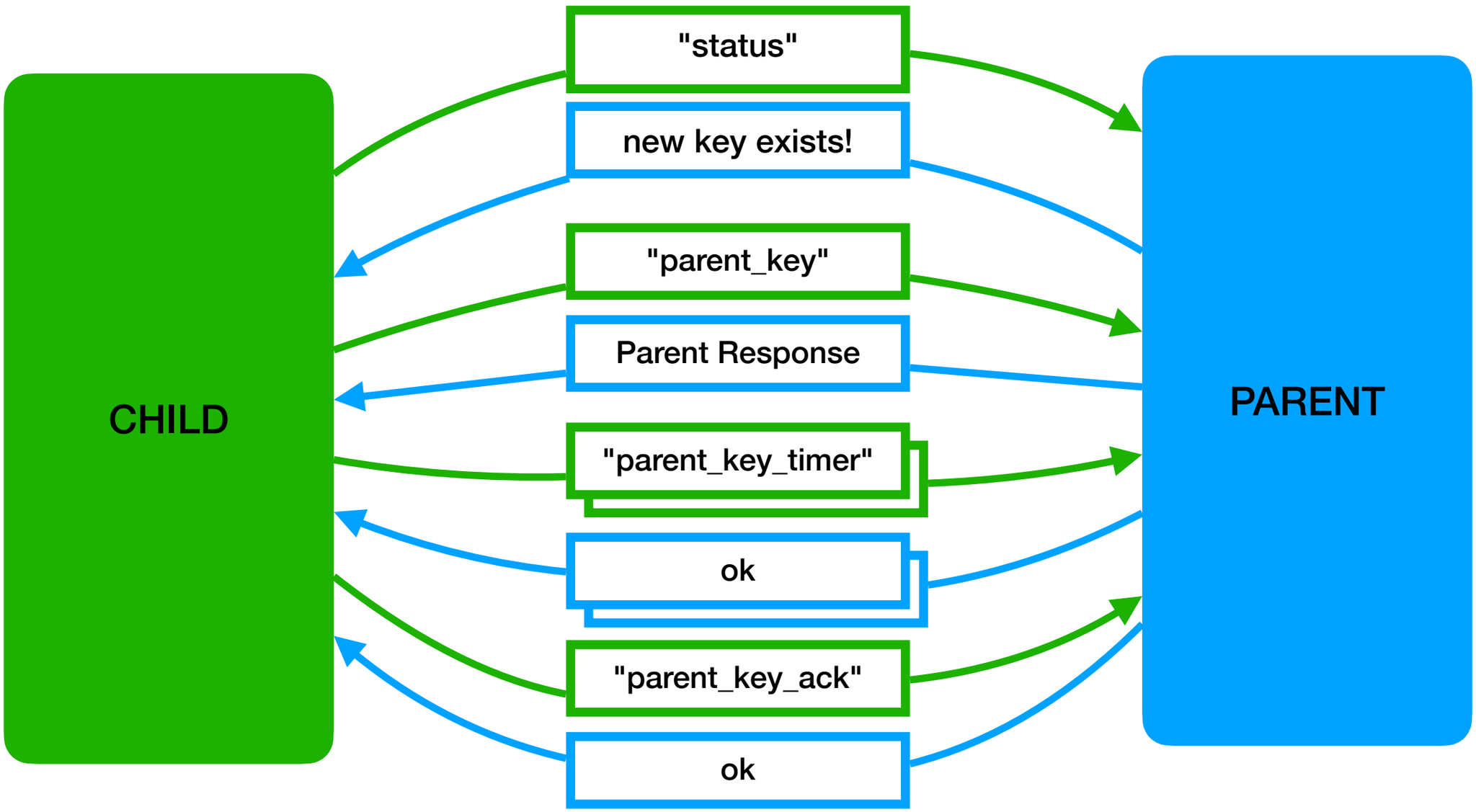


Automated Child Key Roll



Automated Parent Key Roll

Use new exchange instead of "list", let's call it "status". The response is like the RFC 6492 list response, but may include a "new parent key" notification. (use EE cert extension?)



Proposal

Two separate documents:

- **BCP for manual BPKI Key Rollover**
- **Standard for automated BPKI Key Rollover extensions**