

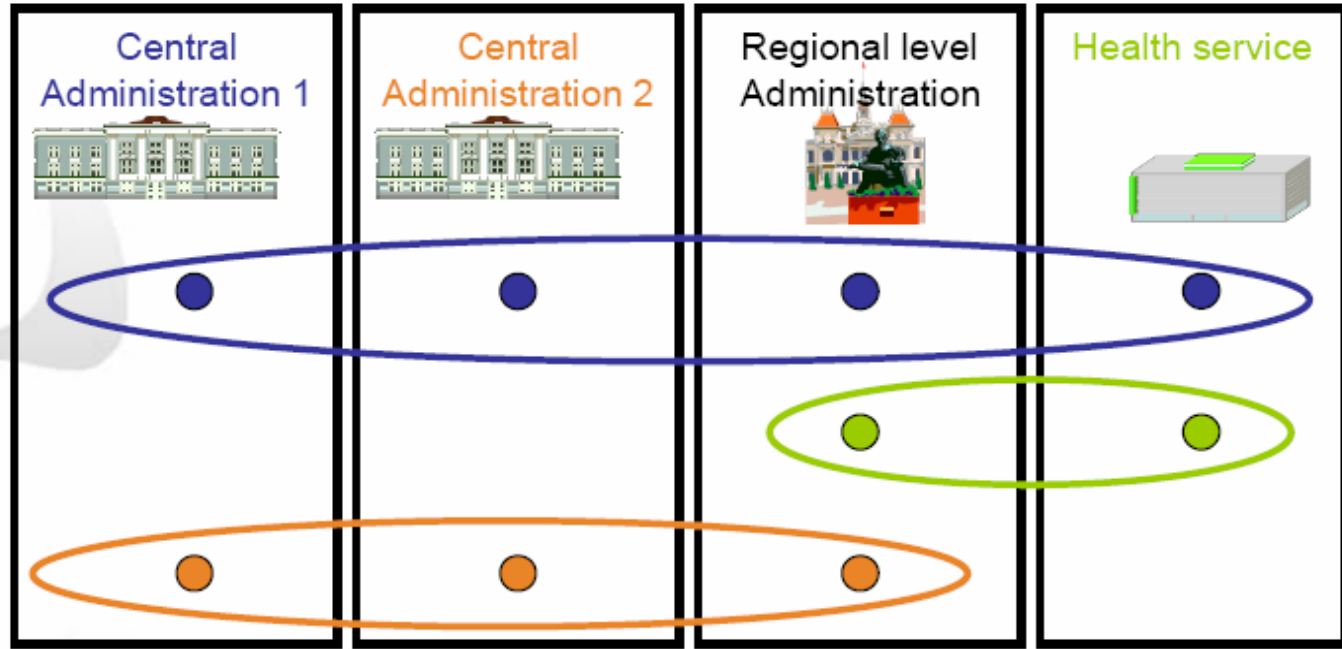
PRESTO

PRotocolle d'Echanges SStandard et Ouvert de l'Administration

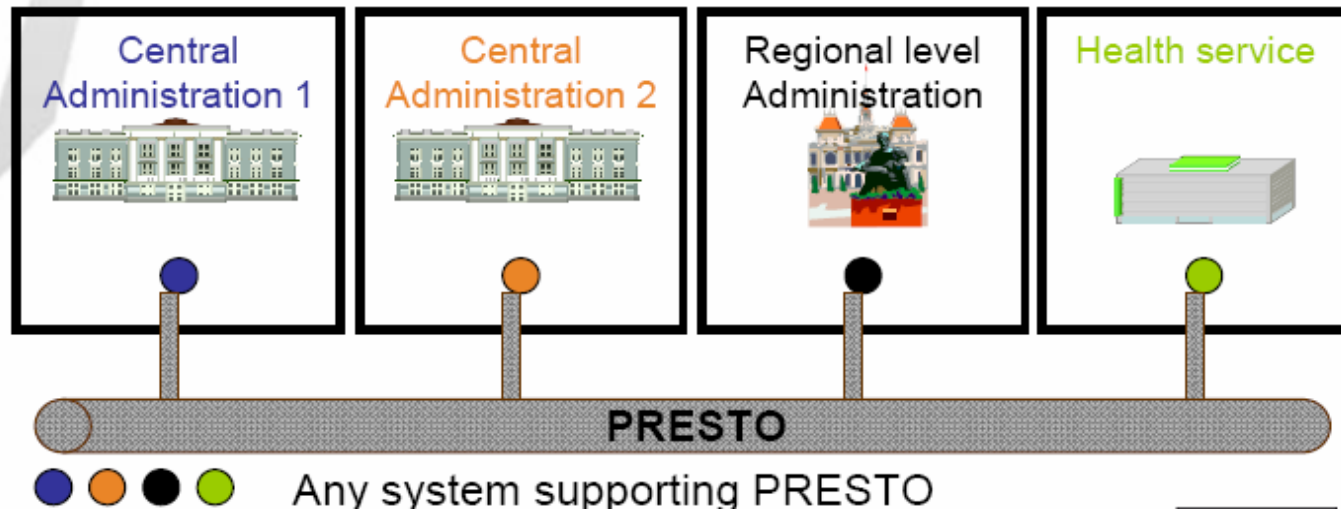


Kontext

Now
Many Systems,
Many protocols



With PRESTO
Many Systems,
One protocol



Ausgangspunkt

- Träger: **DGME** – *Direction Générale de la Modernisation de l'État*
- Evaluierungsdokument zu
 - FAST (*“Fournisseur d'accès sécurisé transactionnel“*, Caisse des dépôts)
 - eLink (IDABC - Europe)
 - ebMS (OASIS)
- Kommentierung zwischen Juni und September 2005
 - von: 9 Unternehmen, 7 öffentliche Einrichtungen, 3 Ministerien, 1 Verband
- Bewertungen
 - FAST ist proprietär
 - eLink ist (noch) nicht operabel (**inzwischen aufgegeben**)
 - ebMS inkompatibel zwischen v.2 und v.3 - und *zu komplex*



⇒ Startschuss für Presto

Am Konzept wesentlich beteiligt:

- Frédéric Law-Dune, DGME
- Maël Ropars, seinerzeit Consultant Unilog, heute BEA
- Philippe Béraud, Microsoft (Beratung zu Web Service Standards)

Wesentlich inspiriert durch:

- WS-RAMP (WS-Profilierung von IBM, Ford und Daimler)
- WS-I Basic Profil, exklusive Nutzung WS-Standards

Stand:

- Dokumente zu Version 1.0 (Juni 2006)
 - Référence technique
 - Guide d'implémentation
- Prototypische Implementierungen Oktober 2006
- 2007 Arbeiten an Version 2.0 begonnen
(Abstimmung mit OSCI, aktuell auch SHS, Schweden; Koordination IDABC)

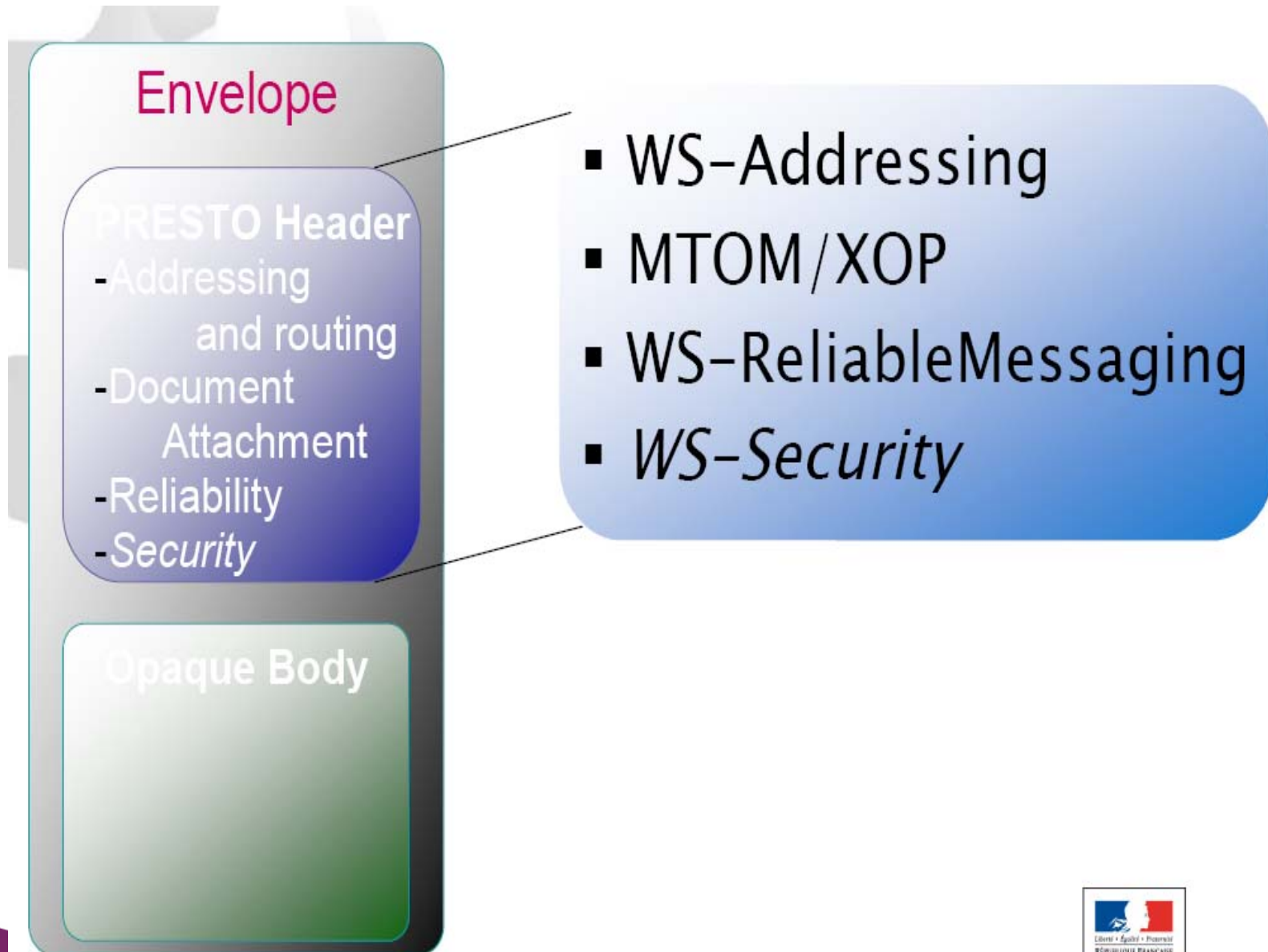


High level requirements (green: comparison to OSCI)

- Policy: simplify exchanges between all eGov actors (Administrative entities, EU, private sector partners) ✓
- Functionalities: (postal mail inspired) common envelope for messages
 - Addressing (routing) ✓
 - Delivery guarantee ✓
 - Reception acknowledgment (non-repudiation) ✓
 - Closed envelope / encrypted document (confidentiality and integrity) ✓
 - Signature (authentication and integrity) ✓
 - File follow-up (process card/log) ✓
- Interoperability: remain agnostic to underlying technical implementations (Protocols HTTP(S), (S)FTP, SMTP..., Standards : XML, WS-*, ebXML..., Technologies : J2EE, .net, LAMP...) ✓
- Application independent ✓



PRESTO protocol at a glance



Protocol description

Data format

– Transported data

- It is necessary to be able to transport attached documents in binary format (e.g. PDF, GIF, JPEG, RTF...) ✓
- Maximum sizes observed of a few hundreds of Mbytes. ✓ (Policy)
- It is not necessary to check the structure of the transported data ✓

– Binary data Transport

- *For memory:*
 - *SOAP with Attachments (SwA) puts binary data in attachments (MIME)*
 - *MTOM (Message Transmission Optimisation Mechanism) and XOP (Xmlbinary Optimised Package) optimize management of binary data by placing in attachment only the necessary data.*
- MTOM/XOP is the favored mode; but SwA still in discussion (for OSCI 2 we decided for MTOM/XOP)



Protocol description

Data format


- Standards for Web Services ✓
 - *WS-Security, WS-Addressing, WS-Policy, WS-Reliability / WS-ReliableMessaging, WS-Eventing / WS-Notification, UDDI, ...* ✓
(question of details)
 - Certains of these standards are already implemented in frameworks, few others still have an uncertain schedule
 - WS-Security and UDDI are the most quoted standards (no UDDI yet, but under consideration)
 - Orientation towards those standards Web Services seems necessary. ✓



Protocol description

Message exchange format

– Transport and Connectivity

- The need for one-to-many was not identified  (OSCI has such requirements – but matter of implementations)
- The exchange protocol must be independent from the transport protocol ✓
- The most commonly used transports protocols are HTTP, SMTP and FTP (OSCI defines http-Binding in V2.0)
- HTTP(s) will be used as the transport protocol between administrations ✓
- The exchange protocol must be independent from the implementation technology. ✓
- The most commonly used implementation technologies are proprietary J2EE, LAMP and .net ✓



Protocol description

Message exchange format

- Asynchronous exchanges
 - Asynchronous mode must be favoured to allow loose coupling between actors ✓
 - Synchronous offered to application, too ✓
- Correlation management
 - The protocol allows the set up of an identifier to send the answer back to the original sender/requester. ✓ (must)
 - Each sender is responsible for the management of the unicity of its technical identifiers ✓ (GUID)
- Service message management ✓
 - Examples :
 - *Connect, Disconnect, SendMessageId, GetMessage, SendMessageList, SendMessage, Subscribe...* (to be specified in detail)
 - Messages must be traceable and their treatment status must be checkable (to be adjusted with «OSCI-Receipts»)



Protocol description

Message exchange format

- Quality of service ✓
 - messages will not be prioritized ✓
 - Senders can request a reception acknowledgement for the final destinee ✓
 - Management of doubles can be considered (?)
 - Senders must be notified/notifiable in case of:
 - message rejection (certificate error...) ✓
 - non delivery (if via an intermediary node) ✓

Security management

- Authentication
 - The protocol will allow authentication data exchange ✓



Protocol description

Security management

– Signatures ✓

- A message can contain multiple signatures, the message signature must be distinct from those of transported documents ✓
- The protocol must allow message signature but under application responsibility. ✓
- In addition to XML-Signature, the XAdES has been recognized as an option to consider ✓ (suggestion for profiling in ISIS-MTT 2.0 has been presented to ISIS-MTT Board)

– Encryption

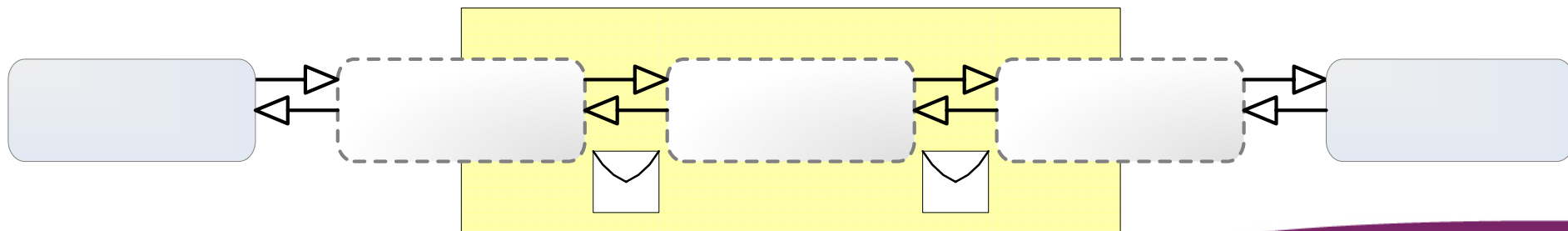
- No encryption mechanism or signing for attached documents is foreseen ✗ (strict requirement for cross-domain scenarios)
- The HTTPS encryption at transport level seems sufficient for minimal security ✓ (OSCI-option instead of «double envelope»)



Proxy description

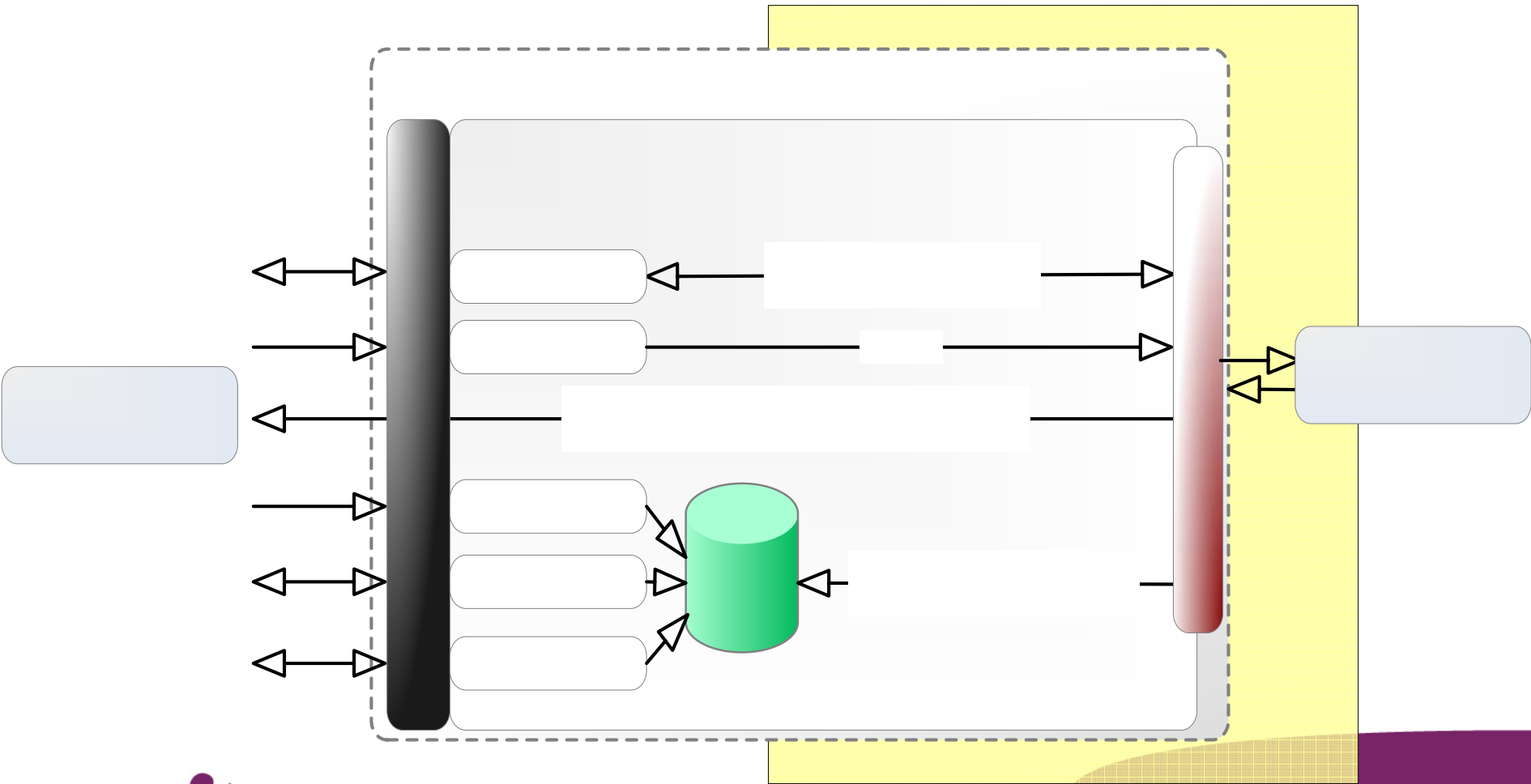
Transport

- In case a exchange hub is used, it will ensure the message persistence (during their treatment by the hub), delivery guarantee and will respect transaction mode. ✓
- The message persistence mechanism can vary depending and the chosen implementation for the hub, either file-based or database-based. ✓
- Difference so far: PRESTO 1.0 knows no «PostBox-Service» (plan: to be adopted from OSCl)



Proxy description

Architectural overview



Industry partners

- **Java technology**
 - **Axway** : use of Systinet Server for Java 6.5
 - **BULL** : use of APACHE Axis 2.0 (OS)
 - **SUN** : use of Java EE 5 (JAX-WS 2.0) and GlassFish/Tango (OS)
- **WCF technology**
 - **Microsoft** : use of .Net Framework 3 (OS)
- **Php technology**
 - **Zend** : use APACHE Axis 2.0 and extensions for php (OS)



Proposal

- PRESTO can be extended to endorse transfer protocol needs between European partners.
- PRESTO is a formidable opportunity to coordinate member States efforts towards an interoperable world.
- We can work together on a European level project to validate the use of PRESTO for inter state exchange



Thank you for your attention!

Jörg Apitzsch, ja@bos-bremen.de

Further Information:

https://www.ateliers.modernisation.gouv.fr/ministeres/projets_adele/a131_b_protocole/public

