



IHE

Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

## **Grundlegende IHE-Profil für Patientenakten**

Workshop: IHE-Anwendbarkeit in der medizinischen Forschung  
Berlin, 2.7.2014



# EHR Implementation Issues

---

- How to format what might be shared?
- How to index what might be shared?
- How to query the index for what might be shared?
- Replicate data to be shared or just manage “links”?
- How to organize “local” sharing vs. “beyond local sharing”?
- How to manage Patient Identity in a sharing environment?
- How to secure the transfer of information?
- How to enable interoperable multi-vendor solutions?
  
- **IHE addresses all those problems**

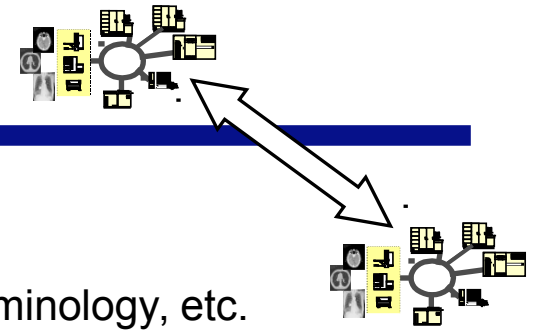


# IHE's Approach to EHR: XDS

- **IHE defines Integration Profiles related to Cross-enterprise Document Sharing**
  - In a wider sense, this family of Integration Profiles usually is subsumed under the label XDS (Cross-enterprise Document Sharing)
  - More specifically XDS is also the name of the core Integration Profile being the base for IHE EHR implementations
- **Why not a single Integration Profile?**
  - Different needs in different projects
  - Different data protection and other laws and regulations
  - Different content to be exchanged
  - Enable step-by-step implementation strategies (extensibility)
- **For a specific project**
  - Select and combine appropriate and required Integration Profiles



# XDS Integration Profiles



- **Base Profile: XDS**
  - Defines basic concepts, actors and transactions, terminology, etc. for cross-enterprise document sharing
  - Permits storing, registering, querying and receiving of medical documents
  - Basically content-neutral: Any type of document can be shared
- **Content Profiles** add rules for specific type of content, for example:
  - XDS-I: DICOM Images
  - XDS-MS: Medical Summaries
  - BPPC: Basic Patient Privacy Consents
  - XPHR: Exchange of Personal Health Record Content
  - XDS-SD: Scanned Documents
  - XD-Lab: Lab Reports
  - PPHP: Pre-procedure History and Physical
  - EDR: Emergency Department Referral

# XDS Support Profiles



- **Several Infrastructure Profiles defined for problems to be solved in typical scenarios**
  - **ATNA: Audit Trail and Node Authentication**
    - Basic security functions
    - *Must* be supported by every XDS-Implementation
    - Centralized audit trail (logging), system authentication  
encrypted transport connections
  - **PIX: Patient Identifier Cross-referencing**
    - Managing multiple local Patient IDs per patient, e. g. for Master Patient Index
    - Look-up service for cross references
    - Support for Master Patient Index (MPI)
  - **PDQ: Patient Demographics Query**
    - Find Patient ID based on name, birth date, sex etc.
  - **CT: Consistent Time**
    - Synchronize all systems to common time
    - Needed for audit trail, access rights etc.

# XDS Support Profiles (2)

---



- **XDR: XDS Reliable Interchange**
  - Point-to-point exchange of clinical documents, e. g. through e-Mail
- **XDM: XDS Media Interchange**
  - Exchange of clinical documents on storage media (CD-R, USB Stick etc.)
- **XCA: Cross-Community Access**
  - Federation of multiple XDS installations
- **XUA: Cross-enterprise User Assertion**
  - User authentication in a distributed system



IHE

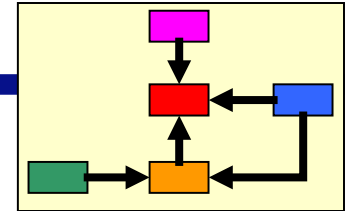
Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

# **XDS – Cross-Enterprise Document Sharing**

*The IHE Core Profile for Building an EHR*

# XDS: Profile Overview



- **Challenges**

- Provide support for document-based patient EHR
- Support for document storage within existing products
- Provide support for indexing of patient documents
- Support query and retrieval of patient documents
- Scalable architecture

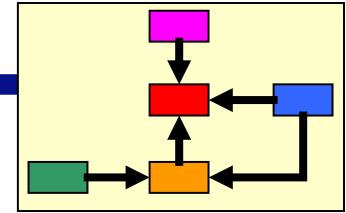
- **Design decisions taken for XDS**

- Documents remain at their originating enterprise (e. g. hospital)
- Documents are indexed by a cross-enterprise registry
- Documents can be searched in registry and received from enterprise
- All enterprises sharing a registry join so-called “Affinity Domain”
  - Group of healthcare enterprises
  - Common set of policies
- Some features optional

- **More details follow, but first we start with the XDS core...**



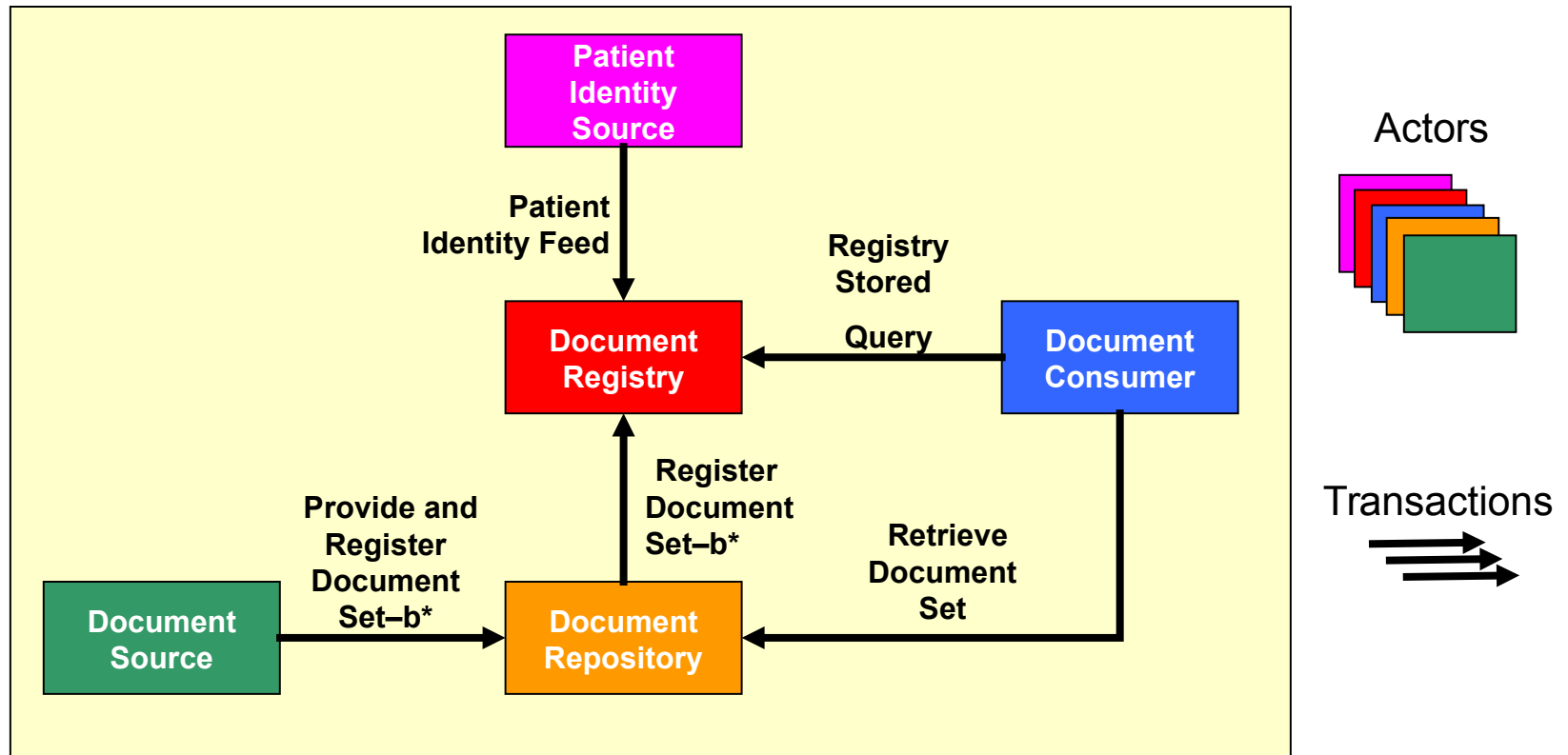
# XDS: Affinity Domain



- XDS Affinity Domain
    - Group of Healthcare Providers working together
    - Goal: Sharing of clinical documents (using XDS)
    - **Organized around a single, central Registry**
  - Actors belonging to the same Affinity Domain have agreed on
    - Document formats to share
    - Common vocabulary to use
    - Common set of Patient Identifiers
    - Common workflow
    - Permission management
    - Security Measures
    - Etc.
- > Setup of Affinity Domain for XDS needs much technical and administrative efforts – more than just buying new servers!

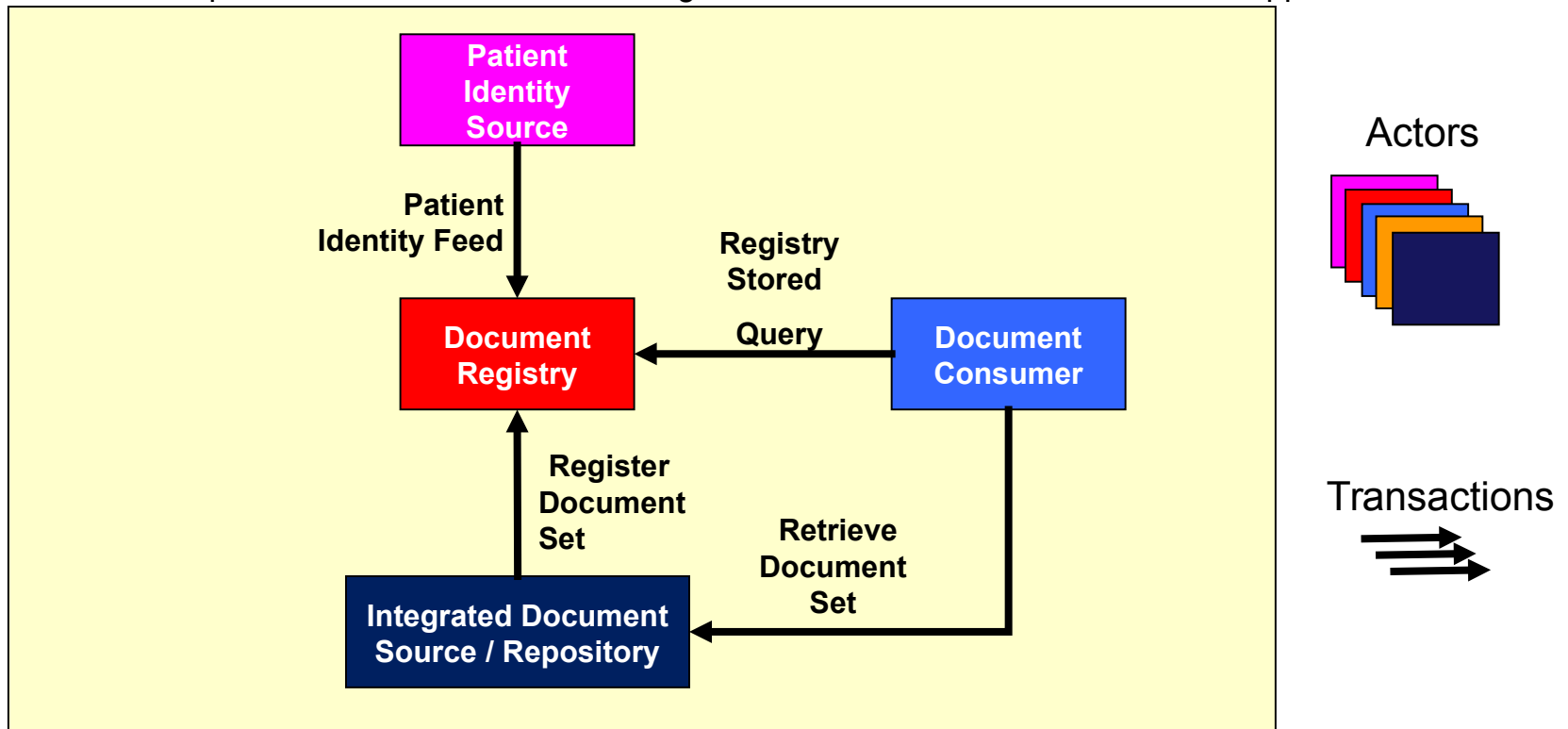
# XDS: Actors and Transactions

- As an Integration Profile, XDS defines Actors and Transactions

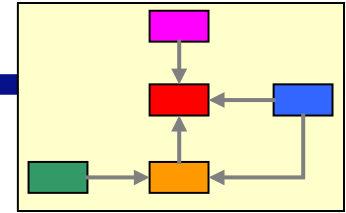


# XDS: Grouping of Document Source / Repository

- Multiple Actors can be grouped, i.e. combined in actual systems, as desired
  - For example, grouping of Document Source and Repository proposed in XDS profile
  - New Actor “Integrated Document Source and Repository”
  - Requirements identical to former single Actors with “internal transactions” dropped out

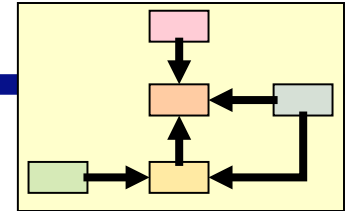


# XDS: Actors



- Functionality of Actors:
  - **Document Source:** Transmits selected documents to Document Repository
  - **Document Repository:** Actually stores documents and registers them at cross-enterprise registry
  - **Document Registry:** Indexes (but not stores!) all shared documents of an Affinity Domain.
    - This actor must be unique in Affinity Domain!
  - **Document Consumer:** Queries registry for documents and receives them (if applicable) from the Repository
  - **Patient Identity Source:** Feeds Registry with patient identifiers and corroborating demographic data like Patients Name, Sex, etc.
- General overview of Actors and Transactions are given in part 1 of the ITI Technical Framework

# XDS: Transactions

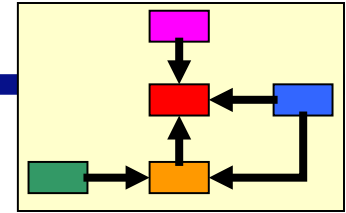


- **Transactions**

- Each transaction identified by identifier “[ITI-X]”, e. g. “Retrieve Document Set” identified by “[ITI-43]”
- Based on existing standards
  - **Patient Identity Feed [ITI-8/ITI-44]** : HL7 V2/V3
  - **Registry Stored Query [ITI-18]**: OASIS ebRIM and ebRS V3.0; different Web Service Standards profiled by WS-I Basic Profile and IHE specifications, e. g. SOAP, MTOM, XOP
  - **Provide and Register Document Set-b [ITI-41]**: Same as for [ITI-18]
  - **Register Document Set-b [ITI-42]**: Same as for [ITI-18]
  - **Retrieve Document Set [ITI-43]**: Same as for [ITI-18]

- Transactions described in part 2a, 2b and 2x (Appendices) of ITI Technical Framework

# XDS: Documents & Submission Requests



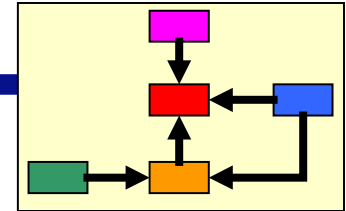
- **XDS Documents**

- Smallest unit of information that can be shared using XDS
- Basically, any type of document supported
  - Handled as binary octet stream originally assembled by Document Source
  - However, no mechanisms defined how to access document content
- Document Source responsible for shared Documents
- E. g. needs to replace Documents which have been submitted with errors
- Either Document status “Approved” (initial status) or “Deprecated”

- **Submission Requests**

- Subsumes transactions “Provide and Register Document Set” and “Register Document Set”, i.e. the collection of information transferred by either of both transactions
- Contains “Submission Set”, i.e. zero or more Documents, Relationships (“Associations”) and Folders
- Processed in atomic manner: Either fully accepted or discarded

# XDS: Folders

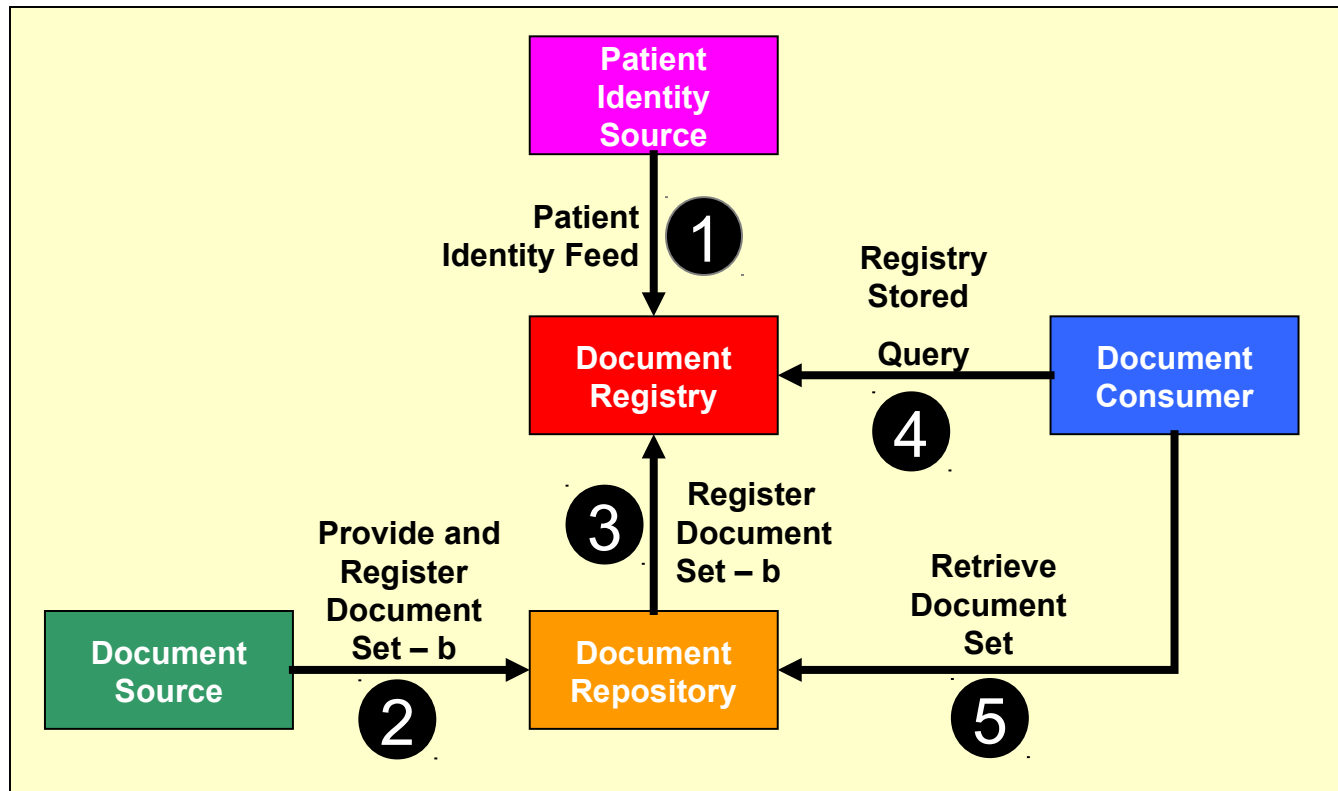


- **XDS Folders**

- Means for grouping XDS Documents
- One title per folder
- Single patient per folder
- May be filled with Documents from different Document Sources
- One Document can reside in more than one folder
- No nesting (i.e. subfolders) permitted
- Semantic is arbitrary: Period of care, dedicated to specific medical problem, etc.

- **Optional for Document Source but required for Registry**

# XDS: Transactions Details

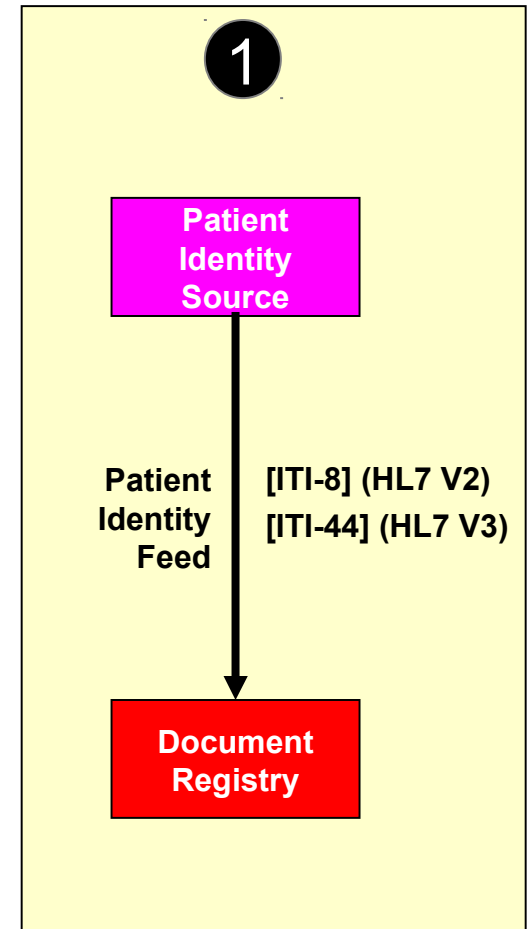


- Let's have a deeper look into the five XDS transactions!

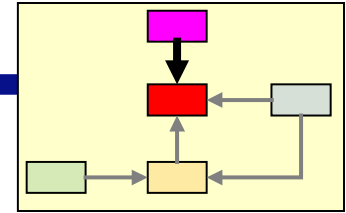


# XDS: Patient Identity Feed

- Each Document linked to a single patient, i.e. to a single patient identifier
- Registry must know patient before accepting any objects (Documents, Folders, Submission Sets, Associations) for it
  - Otherwise Registry rejects documents or queries for that patient
- Patient Identity Source Actor responsible for patient identifier feed
  - Registers new patients or merges existing patients in Registry

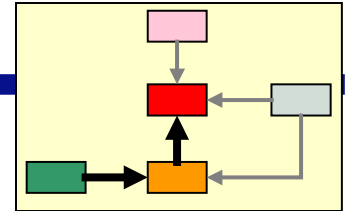


## XDS: Patient Identity Feed (2)

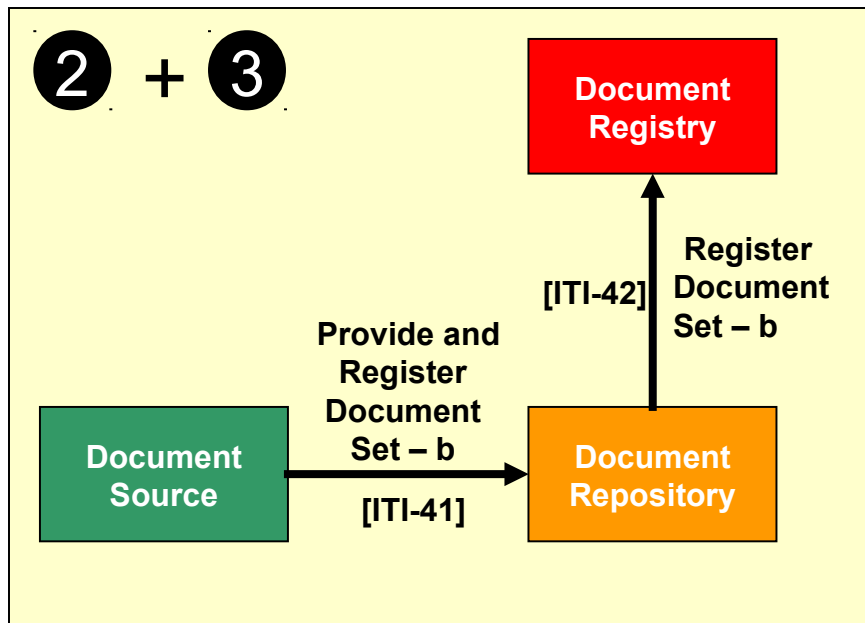


- Either HL7v2, HL7v3 or both must be supported
- Patient Update Messages explicitly not permitted
  - Only messages for “new patient” or “patient merge” available
- Patient Identity Feed not used for establishing “patient database” including patient names etc.
  - Only attribute “Patient Identifier List” must be supported by Registry, other HL7 attributes (e. g. Patient Name) may be ignored!
  - Therefore, demographic attributes not available for search (see “Stored Query” Transactions discussed later)

# XDS: Sharing Documents

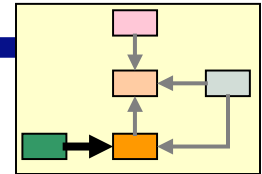


- Two step process for sharing documents in XDS
  1. Document Source *stores* documents to Document Repository
  2. Document Repository *registers* document to Document Registry



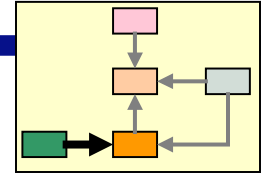
- Document Repository actually stores document
- Document Source *may* also keep document copy
- Document Registry *never* stores actual document but only related metadata

# XDS: Document Metadata



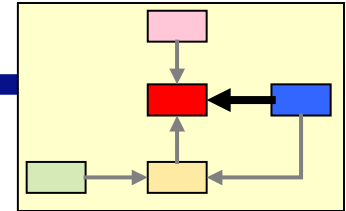
- For each document, metadata is stored in Registry (not complete; either **required**, **required if known**, **optional**)
  - author: author(s) of the documents, with sub-fields institution, person, role, specialty
  - classCode (coded document type, specific to Affinity Domain)
  - confidentialityCode: Code specifying the level of confidentiality of the XDS Document
  - creationTime: Time the author created the document
  - serviceStartTime: Represents the start time the service being documented took place
  - serviceStopTime: Represents the stop time the service being documented took place
  - eventCodeList: List of codes describing the main clinical acts associated with document
  - formatCode: Globally unique code describing the document format.
    - For content where there is dedicated XDS content profiles, formatCode is mostly defined by IHE
  - hash: SHA1 hash of the encoded document as computed by Document Source
  - languageCode: Coded identifier of the document's language
  - mimeType: The MIME type of the document
  - practiceSettingCode: Type of institution that performed the corresponding action
  - size: Size in bytes of the byte stream representing the encoded object
  - title: Title of the document (maximum length of 128 bytes, UTF-8)
  - typeCode: Exact type of document, e. g. Ultrasound Report)
  - comments: Document comments

# XDS: Document Metadata IDs

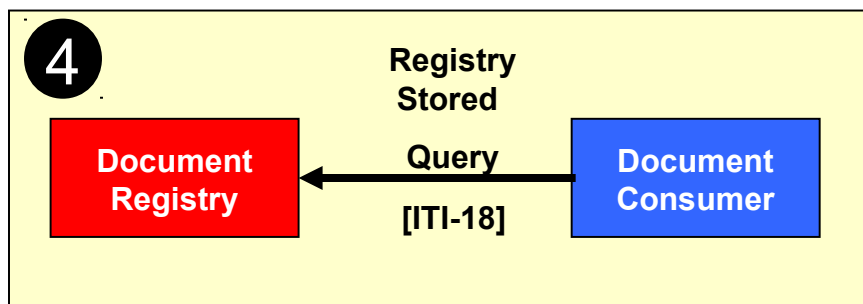


- **IDs identifying the patient belonging to the document**
  - patientId: Authority Domain ID (enforced by the Registry) plus patient ID as used for that patient in Affinity Domain
  - sourcePatientId: Authority Domain ID plus original, “local” patient ID of the patient
- **IDs identifying the document itself**
  - entryUUID: Unique identifier in UUID format within registry, not used in external communication
    - Example UUID format: a6e06ca8-0c75-4064-9e5c-88b9045a96f6
    - (Provide and) Register Document Set transactions may also only provide symbolic (unique) document name. Then, registry has to assign entryUUID itself.
  - uniqueId: Globally unique identifier assigned by the document creator to this document. Used for referencing document in XDS infrastructure.
    - Maximum of 128 bytes,
    - For example: UID taken over from DICOM or for HL7: CDA OID (=UID) plus optional extension
- **IDs referring to the overall XDS infrastructure**
  - repositoryUniqueId: Unique identifier of the Document Repository storing the document in OID format
  - homeCommunityId: ID of the community the document is managed in (based on OID format)
    - Only used in XCA Profile (discussed tomorrow)

# XDS: Registry Stored Query

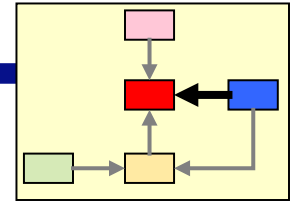


- Permits Document Consumer to query Registry for Document information, including query by
  - Patient identifier, time, type, author
  - Document Source
  - Folder characteristics (time of last update, etc.)
  - Submission set characteristics (submission time, etc.)
- Response contains (if query matches)
  - Meta data for one or more documents, or
  - References to one or more registry objects



- **Queries are predefined by IHE and “stored” in the Registry**
  - Document Consumer selects query and fills in query parameters

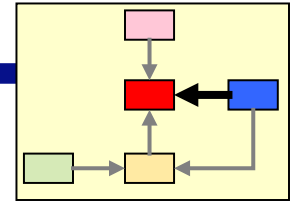
# XDS: Registry Stored Query Types (1)



- **13 different query types**

- **FindDocuments, FindSubmissionSets, FindFolders:** Find Documents, Submission Sets or Folders for a selected Patient ID
- **GetAll:** Get all Registry contents for a selected Patient ID
- **GetDocuments/GetFolders:** Select Document(s) or Folder(s) by their unique identifier(s)
- **GetAssociations:** Get Associations by specifying the source or target (unique identifiers) they link to
- **GetDocumentsAndAssociations:** Get Documents for a selected Patient ID and the Associations linking to them as source/target

## XDS: Registry Stored Query Types (2)

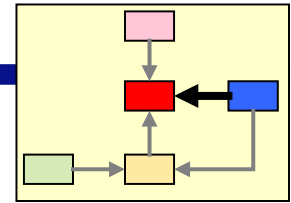


- **Query types (continued)**

- **GetSubmissionSets:** Get a Submission Set belonging to a selected Folder or Document
- **GetSubmissionSetAndContents:** Get Submission Set by its unique identifier and also include
  - Associations contained
  - Folders and Documents targeted by those Associations
  - Associations linking those Documents to Folders
- **GetFolderAndContents:** Get Folder by its unique identifier and also include any contained Associations and Documents they are linking to

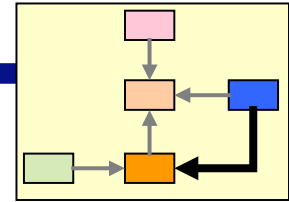


## XDS: Registry Stored Query Types (3)

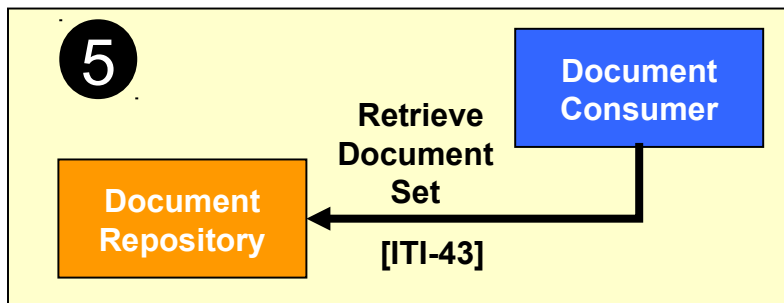


- **Query types (continued)**
  - **GetFoldersForDocument:** Get all Folders that contain a given Document (specified by its unique identifier)
  - **GetRelatedDocuments:** Get Documents related to a given Document (specified by its unique identifier) by Addendum, Replacement or Transformation relationships
- Many queries support further parameters to refine search, e.g. creation time of objects, Document status (“Approved”, “Deprecated”)
- **Each query identified by Universally Unique Identifier (UUID)**
  - E. g. “FindDocuments” identified by “urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d”

# XDS: Retrieve Document Set



- Document Consumer Actor retrieves a set of Documents from Document Repository
  - Consumer must have obtained Document's UUID and responsible Repository from the Registry
  - Repository only knows about Documents (not Folders, Submission Sets, Associations)
- Structure of request message:
  - Web Service (SOAP 1.2) message listing all Documents that should be downloaded



- Response
  - If Documents are found, Repository mainly returns Document Base64-encoded (SOAP with MTOM / XOP)
  - Otherwise, warning or error status can be returned (separately for each requested Document)



IHE

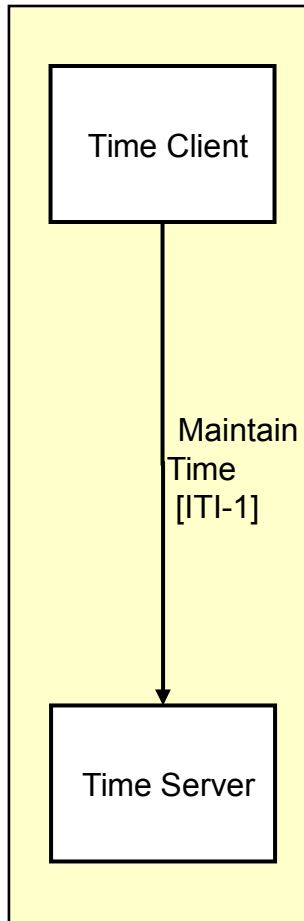
Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

## **CT – Consistent Time**

*Required for all XDS Actors*

# CT: Actors and Transactions



- **Time Client and Time Server “speak” NTP**
  - Client queries Server for UTC time and updates its local system clock after receiving response
  - Complex time estimation algorithms used in NTP
  - Permits synchronization accuracies of milliseconds or even better
- **SNTP (Simple Network Time Protocol)**
  - If accuracies of maximum one second sufficient
  - Permitted if Client and Server are not grouped **AND** one second accuracy is seen as sufficient
  - Optional for Client, not relevant for server (cannot distinguish between SNTP and NTP clients anyway)
- **Secure NTP Option for Client and Server**
  - Cryptographic authentication for NTP
  - Not necessary/recommended for standard scenarios



IHE

Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

## **ATNA – Audit Trail and Node Authentication**

*Required for all XDS Actors*

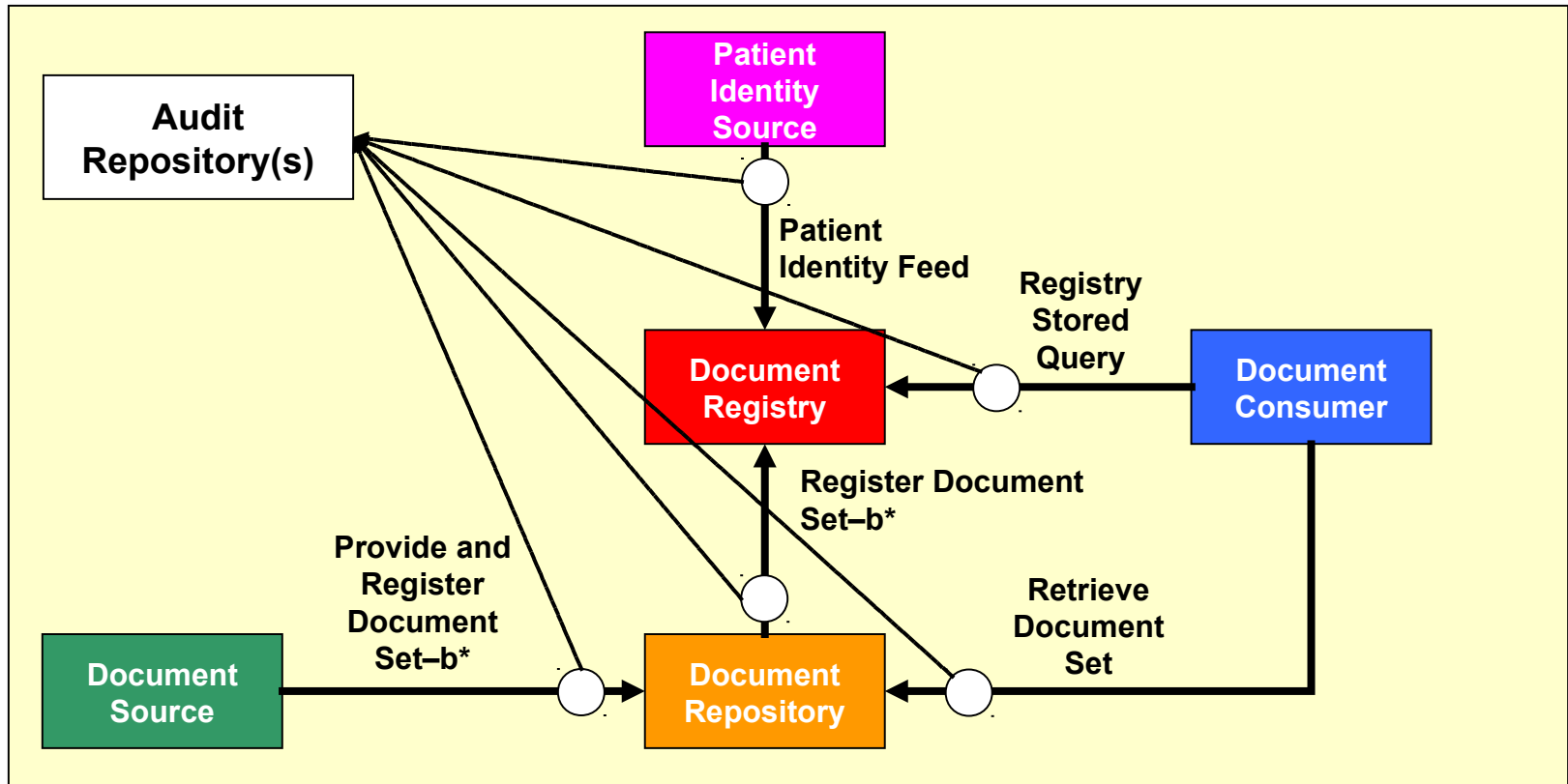


# ATNA: Motivation and Overview

- **ATNA Profile must be supported by every XDS Actor!**
  - Applications become „Secure Applications“ and complete systems „Secure Nodes“
  
- **ATNA Integration Profile requires Actors to provide (if applicable)**
  1. **AT: Audit Trail**
    - Security audit logging to for tracking security events
  2. **NA: Node Authentication**
    - Every machine is host-authenticated, i.e. known to security system of the hospital with known security characteristics
    - Besides other, required for automatic processes running on machine
    - Access to Personal Health Information (PHI) denied for non-authenticated machines (however, exceptions possible)
  3. **User authentication and authorization**
    - No specific technical specifications defined in IHE
  4. **Defined set of security policies**
    - e. g. physical access control, personnel policies



# ATNA: Applying ATNA to XDS scenarios





# ATNA: Authenticate Node Transaction

- XDS Actors (being Secure Nodes) need to authenticate with each other
  - Using certificate-based system authentication
  - For all relevant XDS protocols possible:  
TLS-based encryption defined for DICOM, HL7 and HTTP/SOAP
  - User authentication addressed by separate Profiles (XUA/EUA) discussed later
- Encryption of transmitted data permitted but not enforced by ATNA
  - Usually already implemented in hospital by VPN or other means
  - Optional for all ATNA Actors (Option “ATNA Encryption”)
    - Requires at least TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA profile support
- Access denied for non-authenticating hosts
  - Or guaranteed that no access to PHI possible





IHE

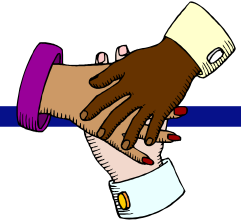
Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

## **PIX – Cross-referencing Patient Identifiers**

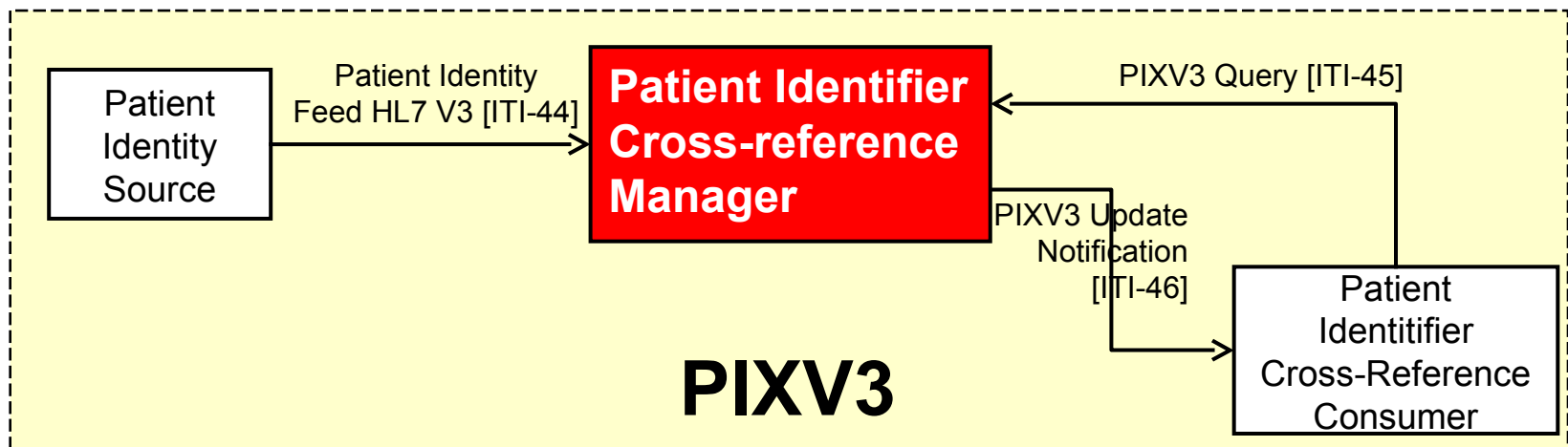
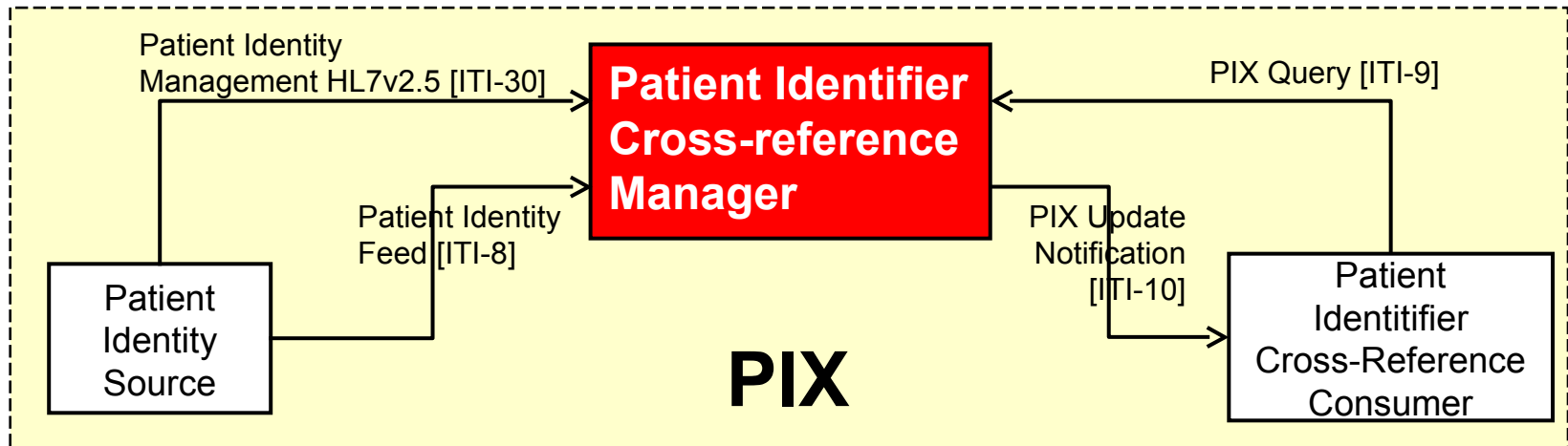
*Optional XDS Infrastructure Profile*

# PIX: Motivation and Concept

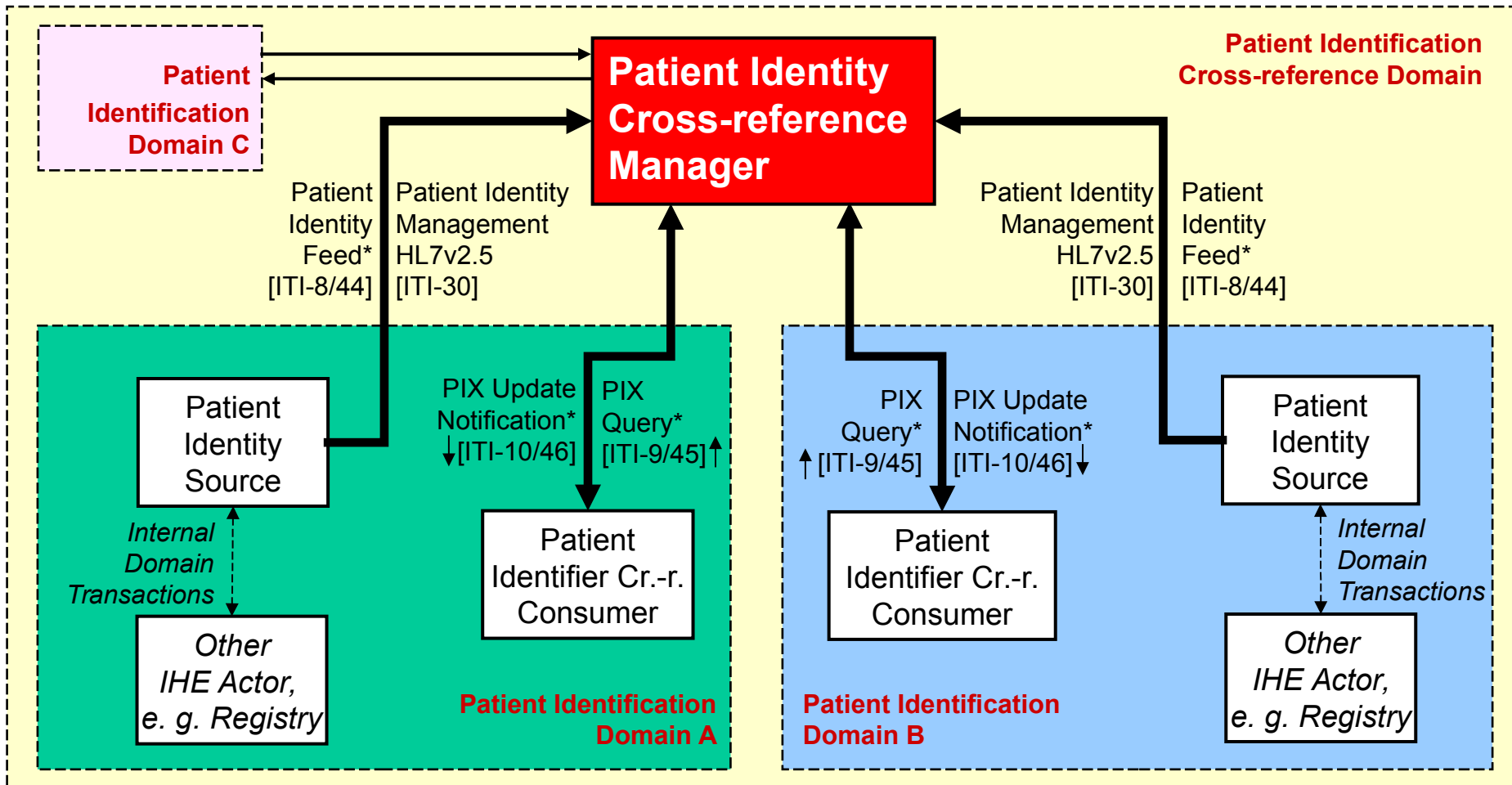


- **In many scenarios different patient identifiers (patient IDs) co-exist for a single patient**
  - Different enterprises in an XDS Affinity Domain probably have their own patient IDs, i.e. they are part of different Patient Identifier Domains
  - Even in a single enterprise, different departments may use different Patient Identifier Domains, e.g. radiology and cardiology department
  - XDS requires all Actors to use a common Patient Identifier Domain for all document submissions and queries.
- **Idea: Establish system to keep track of all patient identifiers**
  - Fed with patient IDs by systems originated in different Patient Identifier Domains
  - Can be queried for all or selected patient IDs belonging to a specific patient
- **The PIX Integration Profiles utilizes above approach**
  - PIX stands for “Patient Identifier Cross-referencing”
  - Defines Actors “Patient Identity Source”, “Patient Identifier Cross-reference Manager” and “Patient Identifier Cross-reference Consumer” and the corresponding Transactions
  - Two profiles (PIX and PIXv3), at least principally identical

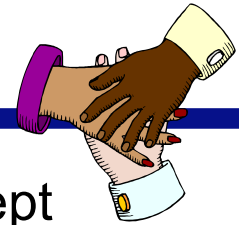
# PIX and PIXV3: Actors and Transactions



# PIX and PIXV3: Actors and Transactions Example



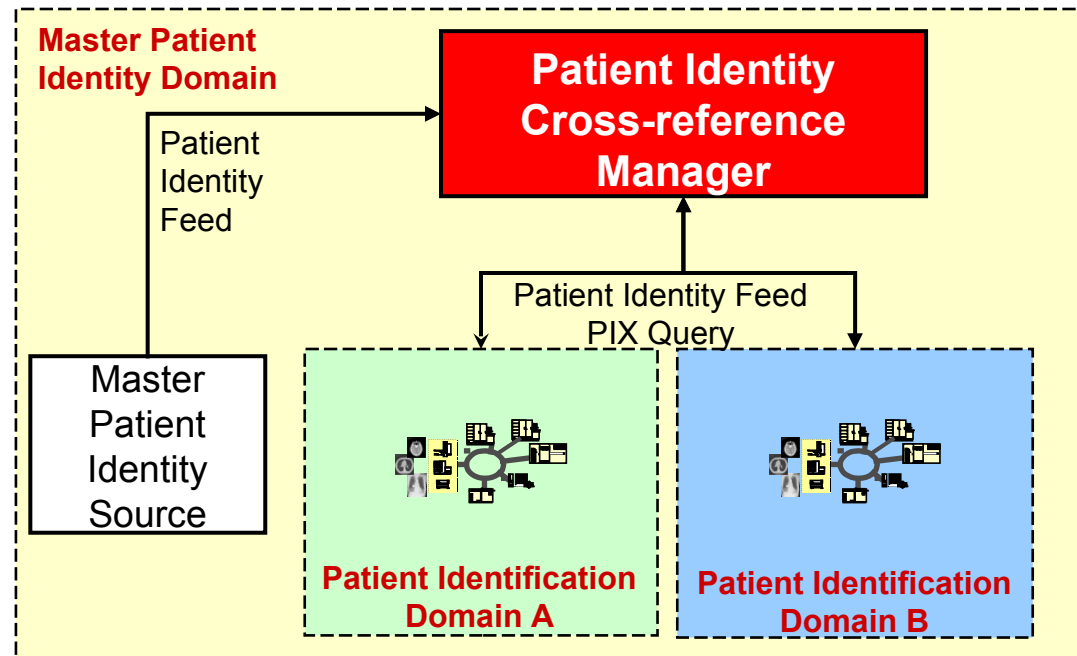
# PIX: Master Patient Index



- Master Patient Index (MPI) usually understood as concept of introducing a “top-level” patient ID per patient
  - i. e. establishing a Master Patient Identity Domain
  - Straightforward with PIX

MPI example scenario

- Slightly different approaches possible, for example
  - Dedicated Master Patient Identity Domain (may only consist of Patient Identity Feed)
  - Existing Patient Identity Domain declared as Master Domain





IHE

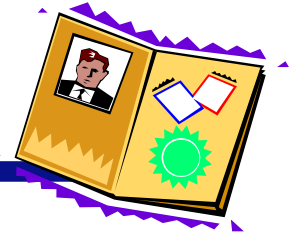
Integrating  
the Healthcare  
Enterprise

*Michael Onken, onken@offis.de  
IHE Technical Manager Deutschland*

## **PDQ – Patient Demographics Query**

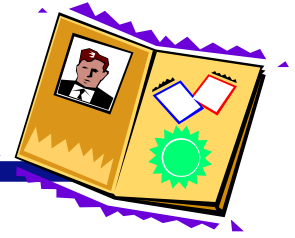
*Optional XDS Infrastructure Profile*

# PDQ: Motivation and Goals

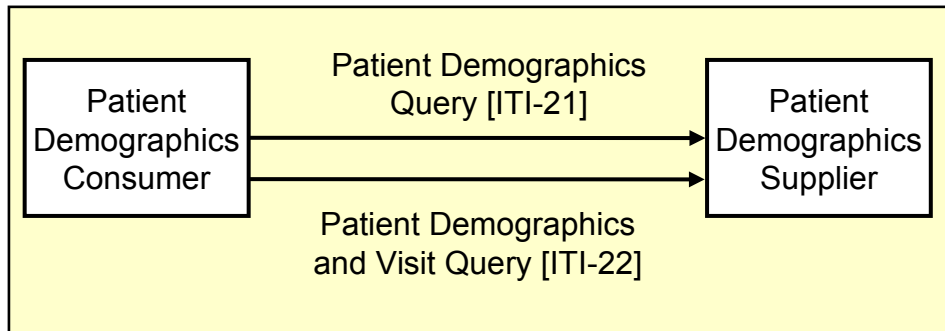


- **Searching in a patient database is a common task**
  - Nearly any system in hospital needs access to patient information, e. g. bedside systems, physician office systems, lab applications
  - Distributed, isolated patient databases lead to duplication and inconsistency of patient data
  - Often only partial patient information available
- **Idea: Establish central, searchable patient database**
- **PDQ Integration Profile addresses these issues**
  - PDQ stands for “Patient Demographic Query”
  - Permits query for full demographic data using (minimal) demographic data
    - Thus, “broader” concept compared to PIX only permitting access to patient IDs
  - Defines Actors “Patient Demographics Supplier” and “Patient Demographics Consumer” as well as the corresponding Transactions
  - Can manage multiple patient identifiers from different domains for a single patient

# PDQ: Actors and Transaction

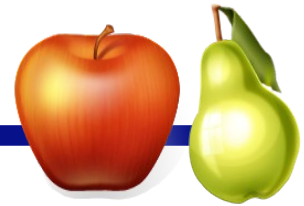


- Only two Actors involved, using one required, one optional Transaction
- Actor “Patient Demographics Supplier”
  - Holds database with demographics and optionally visit information for a patient
  - No rules in IHE on how to fill this database (e. g. by other IHE Actor)
  - May be established within single Enterprise or Cross-Enterprise, i. e. for XDS Affinity Domain
  - Shall return at least all exact matches, but may send others (e. g. phonetic matching, other heuristics)



- Actor “Patient Demographic Consumer”
  - Requests patient information based on known demographic data
  - Receives list of patients matching the query attributes





# PDQ and PIX: Comparison

- **PDQ and PIX serve similar purpose**
  - Distribution of patient information cross departments or enterprises
    - Within single identifier domain, PIX does not make much sense
- **But some essential differences**
  - PIX only for patient identifiers with demographic data only used as corroborating information for linking in Cross-reference Manager
    - PDQ also publishes demographic information for query and download
  - PIX includes feeding of patient database by specific Actor/Transaction
- **PIX versus PDQ: Which to choose for XDS?**
  - Both possible, if management of patient identifiers is main purpose
  - PIX, if querying systems not interested in demographics but only in identifiers
  - PDQ, if distribution of demographic data also targeted



**IHE**

Integrating  
the Healthcare  
Enterprise

*Michael Onken, [onken@offis.de](mailto:onken@offis.de)  
IHE Technical Manager Deutschland*

**Thank You for Your Attention!**

