



CATCHPlus and EPIC: an update

Hennie Brugman

Technical coordinator CATCHPlus

Meertens Institute



Summary

- CATCH & CATCHPlus
- Initial requirements from CATCHPlus and CH
- Progress report
 - Base technology
 - Identifier management (API, application case)
 - Organisational embedding
- Lessons learned, open issues
- Plans
- Concluding remarks

CATCH



National Archive

National Library of the Netherlands

Netherlands Institute for Sound and Vision

Gemeentemuseum Den Haag

Rotterdam Municipal Archives

Naturalis (National Museum of Natural History)

- CATCH Rijksdienst voor het Cultureel Erfgoed

- CATCH Meertens Institute

- 8 subprojects at large CH institutions

- Connected by common services

- Vocabularies, Workspaces, Annotations, User Profiles

- Infrastructural: OAI-PMH, **persistent identifiers**

- Project bureau hosted by Meertens Institute

- www.catchplus.nl



Initial requirements from CATCHPlus and Cultural Heritage

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Requirements (1)

Software support

- Good resolving service available
- Proven technology, stable and 100% reliable
- Scalable, with respect to
 - Number of identifiers
 - Performance
- Globally working solution
- Distributed hosting and service providing possible
- Identification of parts of objects
- Possibility to associate metadata with an identifier
- “Actionable”: identifiers can be resolved using http URI

Requirements (2)

Identifier management

- Identifier management should be independent of
 - System management
 - Web server management
 - Hosting of resolution services
- Can be done from the context of a collection management system
 - typically by a responsible collection manager
- Is efficient, powerful and simple
- Is secure

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Requirements (3)

Organisation, policy

- What choices are made by partner institutions ?
(the fewer 'flavours', the better)
- Reliability and sustainability of the service providers
- Quality of Service: redundancy, high availability, performance, capacity to scale up
- Limited and controlable costs
- Freedom to switch between service providers
 - Delegation of PID administration
- Control by user community



Where are we today?

CATCH



CONTI
ACCES
TO
CULTU
HERIT/
PLUS

Requ
S

Local Handle Systems

- 1 per participating Naming Authority
- Hosted by SARA

- ✓ Good resolving service available
- ✓ Proven technology, stable and 100% reliable
- ✓ Scalable, with respect to
 - ✓ Number of identifiers
 - ✓ Performance
- ✓ Globally working solution
 - Distributed hosting and service providing possible
 - Identification of parts of objects
- ✓ Possibility to associate metadata with an identifier
- ✓ “Actionable”: identifiers can be resolved using http URI

CATCH



CONTINUOUS
ACCESS
TO
CULTU
HERIT/
PLUS

• Mirrored by EPIC

Requirements (1)

Software support

- ✓ Good resolving service available
- ✓ Proven technology, stable and 100% reliable
- ✓ Scalable, with respect to
 - ✓ Number of identifiers
 - ✓ Performance
- ✓ Globally working solution
- ✓ Distributed hosting and service providing possible
- Identification of parts of objects
- ✓ Possibility to associate metadata with an identifier
- ✓ “Actionable”: identifiers can be resolved using http URI

CATCH



CONTINUOUS
ACCESS
TO
CULTU
HERIT/
PLUS

• Requires Handle software update
by SARA/EPIC?

Requirements (1)

Software support

- ✓ Good resolving service available
- ✓ Proven technology, stable and 100% reliable
- ✓ Scalable, with respect to
 - ✓ Number of identifiers
 - ✓ Performance
- ✓ Globally working solution
- ✓ Distributed hosting and service providing possible
- • Identification of parts of objects
- ✓ Possibility to associate metadata with an identifier
- ✓ “Actionable”: identifiers can be resolved using http URI

CATCH

CATCHPlus RESTful web service

- For searching, creation and management of Handles
 - SARA has built the first version for CATCHPlus
 - Currently operational
 - Available as Open source
-
- ✓ Identifier management should be independent of
 - System management
 - Web server management
 - Hosting of resolution services
 - Can be done from the context of a collection management system
 - typically by a responsible collection manager
 - ✓ Is efficient, powerful and simple
 - ✓ Is secure

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

- Side effect of collection management
- ✓ Side effect of collection publication

Requirements (2)

Identifier management

- ✓ Identifier management should be independent of
 - System management
 - Web server management
 - Hosting of resolution services
- ➡ • Can be done from the context of a collection management system
 - typically by a responsible collection manager
- ✓ Is efficient, powerful and simple
- ✓ Is secure



Case: publishing Handles using OAI-PMH

- Sound and Vision
 - Created and published 1.4 million Handles
- Collection metadata published or updated using **inbox** mechanism
- Dropping in inbox triggers
 1. Update local indexes of OAI-PMH data provider
 2. Corresponding PID management service calls
- OAI data provider also publishes handles for harvesting

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Requirements (3)

Organisation, policy

- What choices are made by partner institutions ?
(the fewer 'flavours', the better)
- Reliability and sustainability of the service providers
- Quality of Service: redundancy, high availability, performance, capacity to scale up
- Limited and controlable costs
- Freedom to switch between service providers
 - Delegation of PID administration
- Control by user community



CATCHPlus solution: base technology

- Based on Handle technology
 - Best meets our requirements (by far)
- One Local Handle System and Handle prefix per participating Naming Authority
- Currently hosted by SARA, eventually mirrored by other EPIC partners (redundant hosting)
- For part identifiers: interested in testing and using template handles



CATCHPlus solution: identifier management

CATCHPlus RESTful web service

- For resolving, searching, creation and management of Handles (in one's own Naming Authority) over internet
- Also will support batch operations (“move collection”)
- SARA has built the first version for CATCHPlus
- Available as Open source
- Ideally: merge into a uniform redundant service provided by EPIC

Web GUI will be developed (Q2-3, 2010)

- Prototype for evaluation by collection managers



CATCHPlus solution: organisational embedding

- **EPIC**

- Via SARA (Netherlands) as primary partner
- Hope to arrange via EPIC:
 - Quality of Service
 - Persistence of hosting and services
 - Affordable, non-commercial business model for potentially large numbers of handles
 - Freedom to switch between (primary) service providers
 - Representation of interests of (Dutch) digital cultural heritage (governance structure?)
 - Uniform REST service for identifier management
 - Extensible
 - Community policies via profiles
 - One base URL that efficiently resolves actionable PIDs (<http://hdl.handle.net>)



Application to Cultural Heritage data sets



Collections and data sets

Currently assigning identifiers to:

- Concepts for the CATCHPlus Vocabulary Repository
- A subcollection of the Sound and Vision AV archive's metadata catalog

Several Dutch cultural heritage institutions and projects expressed serious interest

- National Archive, Naturalis, WieWasWie



Publication of PIDs

- When you publish data or metadata, always publish PIDs
- Methods of publication
 - Web pages
 - clickable links for actionable PIDs
 - OAI-PMH metadata harvesting
 - merge PIDs and metadata in your OAI data provider
 - Linked Open Data
 - use actionable PIDs as your resolvable http URIs

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Methods for identifier management

- Side effect of collection management
 - Action in collection management system triggers PID management REST call
- Side effect of collection publication
 - Update of your public data set (e.g. OAI data provider's internal database) triggers PID management REST call



Concluding remarks

- We are quite optimistic about our choice for Handles/EPIC
- But, there is a substantial list of requests to EPIC
- Crucial is good software infrastructure for automatic support for
 - Introduction of PIDs
 - PID management
 - Publication of PIDs
- Web service API is essential part of this software infrastructure



Questions?

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Application to data sets

Some questions to answer first...

- What are the objects to assign persistent identifiers to? (versions, metadata records, formats, composite objects...)
- Is there a relation with already existing identifiers?
- What syntax to use? Include semantics in your PIDs?
- Where do your PIDs resolve to, especially for objects that do not have a web representation of their own?
- Who is responsible for identifier creation and management?
- What guarantees can be made with regard to persistence?
- Who does hosting? Who provides services?



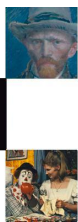
Steps

- For existing objects
 - Determine your policies
 - Determine what URLs to resolve to
 - Create and publish PIDs for these URLs
 - Locally store association of URLs and proprietary identifiers
 - For all externally visible metadata: replace proprietary identifiers with PIDs
- For new objects
 - Ultimately, integrate PID creation and management in your collection management tools and workflows

Sound and Vision pilot

- Objects:
 - metadata descriptions at level of broadcasts
 - Open data set: ‘polygoon journal’
- Existing identifiers: “task identifiers”
- Resolve to metadata record implies: resolve to dynamically created html page
- Persistent identifiers are published using OAI-PMH
 - Published metadata refers back to same dynamic web page
 - OAI data provider uses PID service to find handles for internal identifiers/URLs

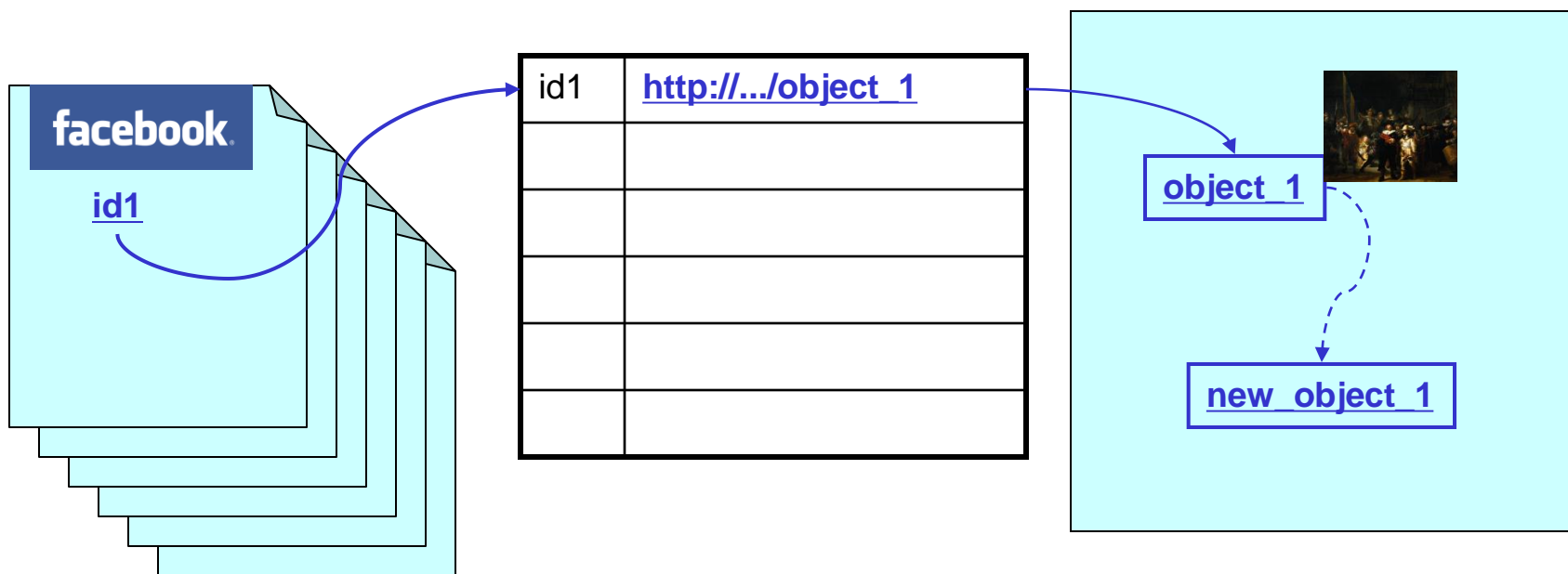
CATCH



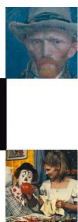
CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS



Basisoplossing



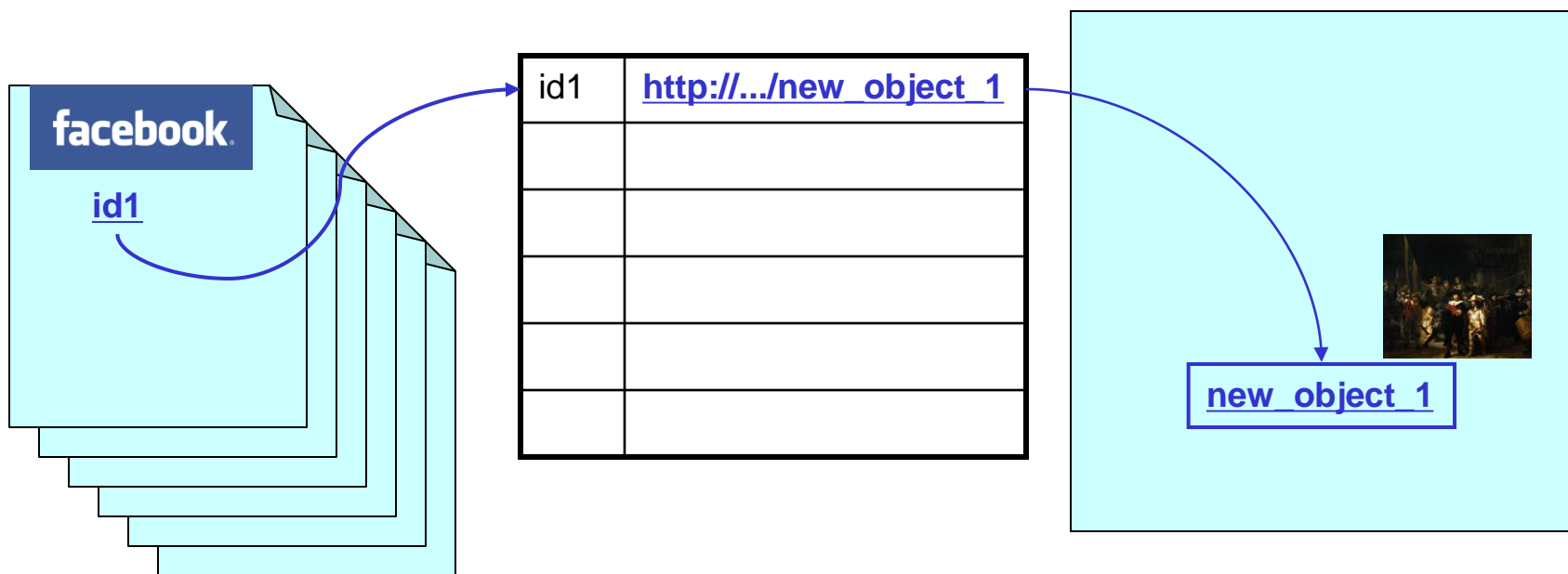
CATCH



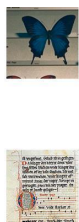
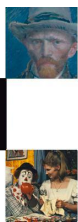
CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS



Basisoplossing

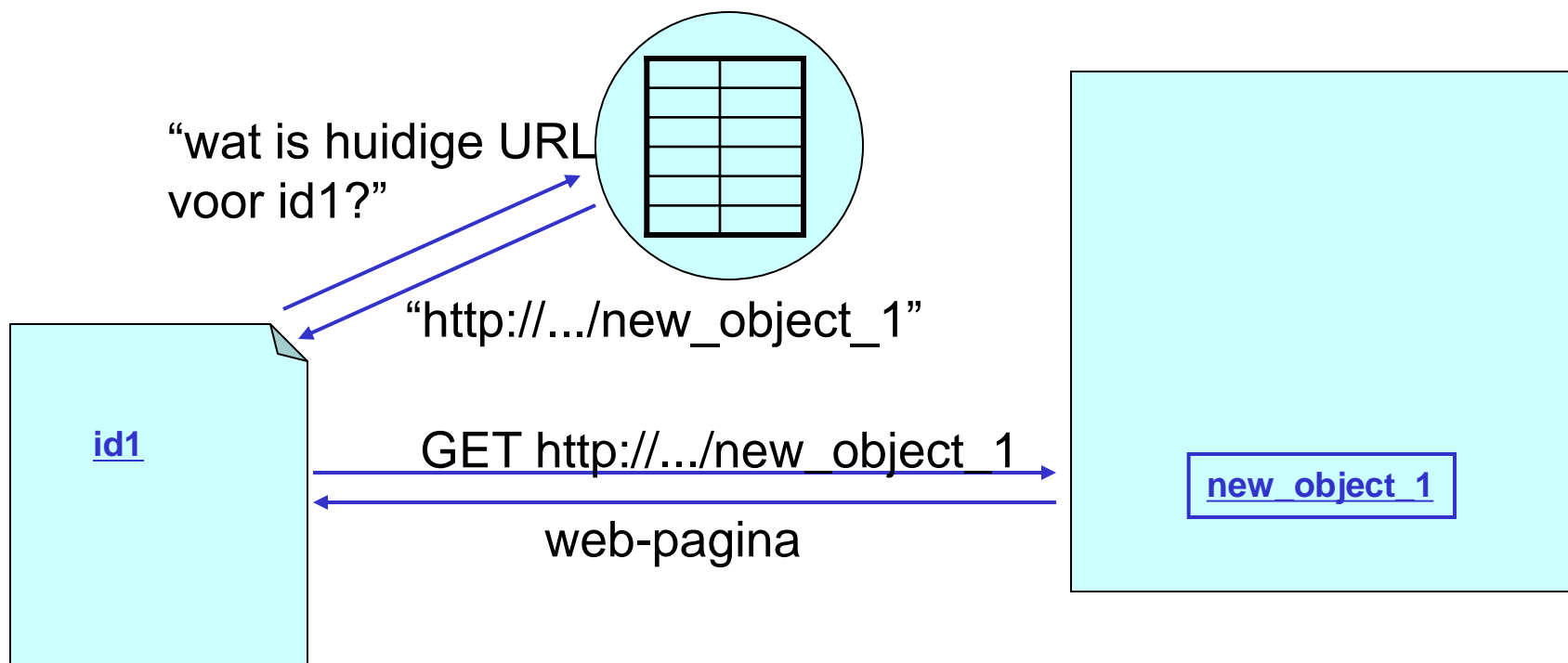


CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Resolver dienst



CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Naming Authority

- Naming Authority: beheerder van de inhoud van de tabel
- NA heeft ook een unieke persistente identifier
- Globaal geregistreerd waar de resolver voor de NA is te vinden.
- Veel soorten persistente identifiers hebben de basisvorm:

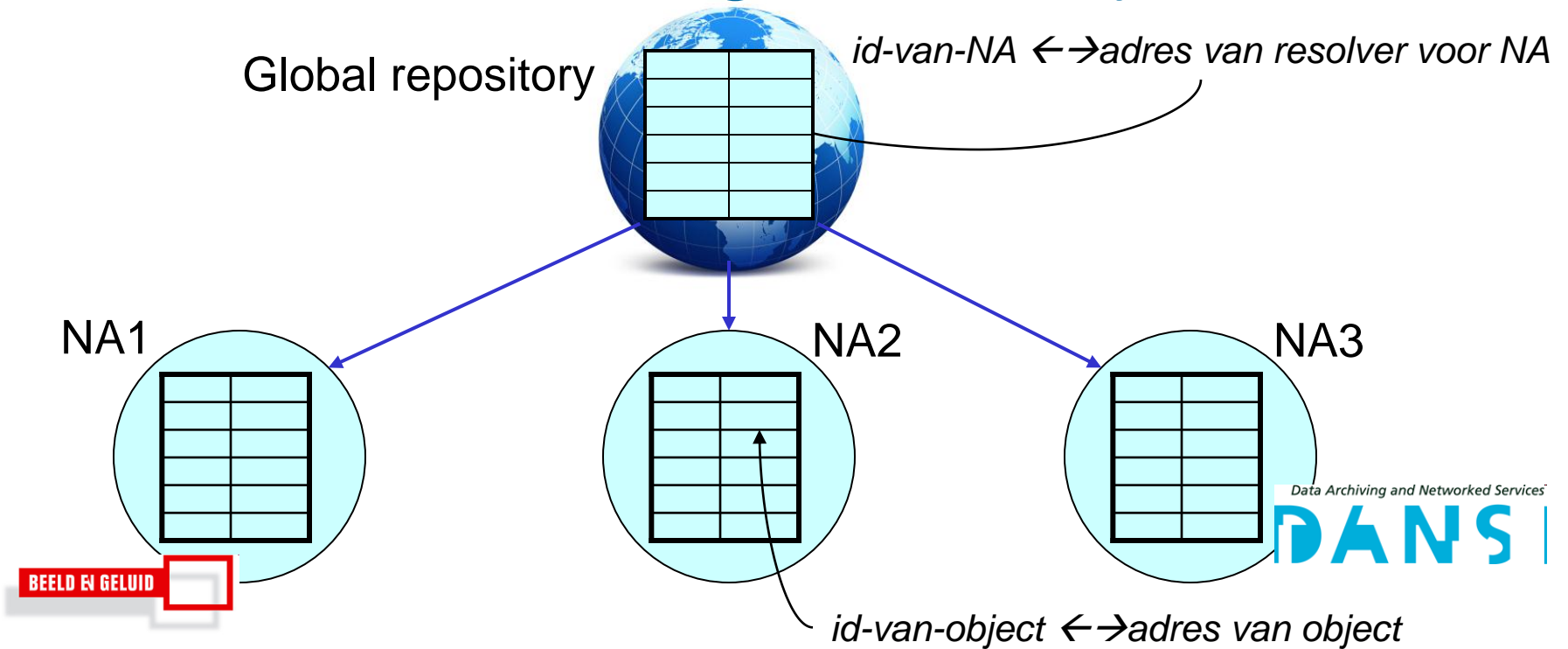
<unieke-id-van-NA><unieke-locale-id>

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Naming Authority

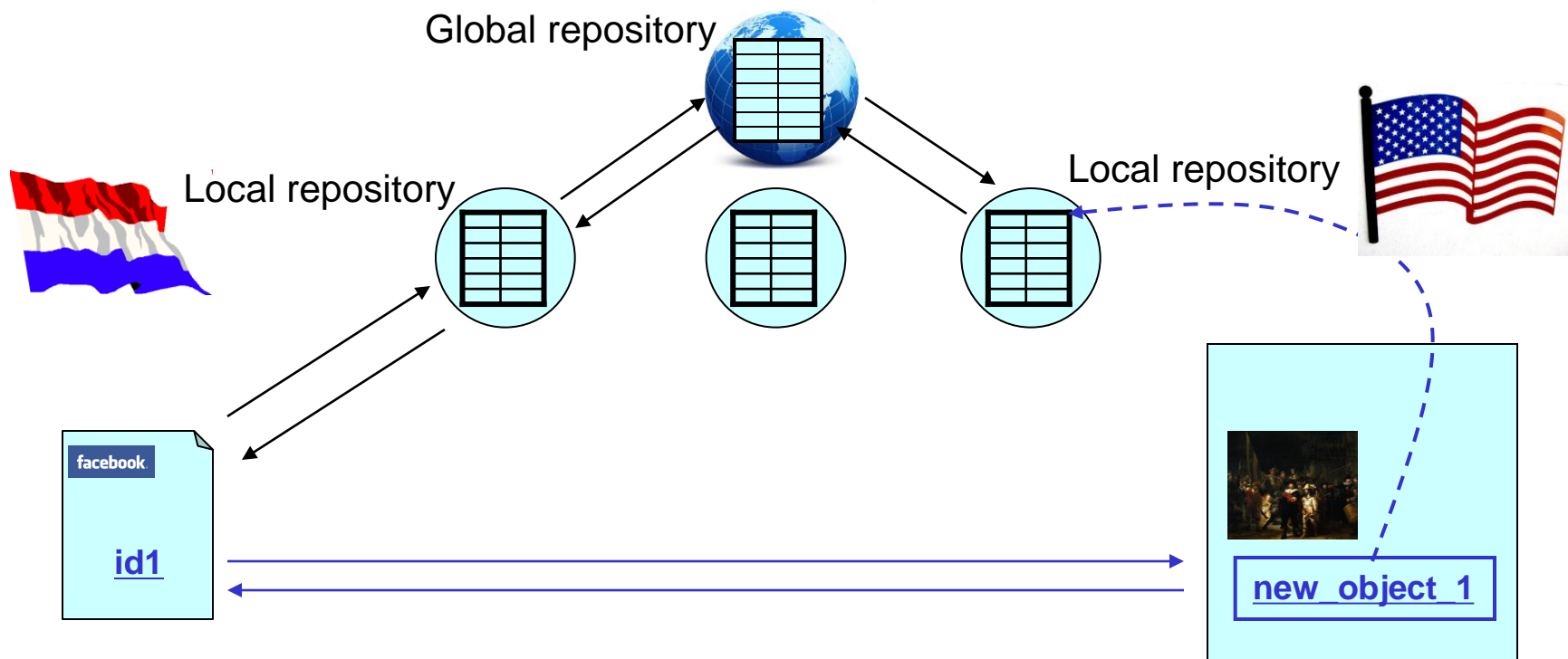


CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

Global resolving





Voorbeelden van PID oplossingen

- Er bestaan een aantal verschillende oplossingen
- Binnen erfgoed in NL spelen een rol
 - Gewoon URLs gebruiken
 - URN-NBN
 - Handles
 - DOI
 - ARK
 - PURL



http URL

- http://identifiers.erfgoed.nl/local_id_1821

URN-NBN

- urn:nbn:nl-local_id_1821

Handles

- 10574/local_id_1821
- http://hdl.handle.net/10574/local_id_1821

DOI

- <doi:10.1594/PANGAEA.726855>
- <http://dx.doi.org/10.1594/PANGAEA.726855>

ARK

- http://identifiers.erfgoed.nl/ark:/128014/local_id_1821

PURL

- <http://purl.org/vocabularies/iconclass/concept1821>

CATCH



CONTINUOUS
ACCESS
TO
CULTURAL
HERITAGE
PLUS

HOST

X

http URL

- http://identifiers.erfgoed.nl/local_id_1821

NA

X

URN-NBN

- urn:nbn:nl-local_id_1821

LOCAL ID

Handles

- 10574/local_id_1821
- http://hdl.handle.net/10574/local_id_1821

DOI

- <doi:10.1594/PANGAEA.726855>
- <http://dx.doi.org/10.1594/PANGAEA.726855>

ARK

- http://identifiers.erfgoed.nl/ark:/128014/local_id_1821

X

PURL

- <http://purl.org/vocabularies/iconclass/concept1821>

- Part identifiers - some use cases
- Selections from audio-visual objects
- Thesaurus concepts
- Lexicon entries

- How: rewrite rules per Naming Authority and/or per PID, built into the resolver

- “*Part identifiers*” preferred over “*many PIDs*” when:
 - potentially infinite number of pids (AV)
 - impractically large numbers of PIDs ($\sim 10^7$ concepts)

- Possible criterium:
 - Managed as one coherent (web) resource (1 base URL)