Background on Namespaces, XML, and Their Applicability to ISO/IEC 11179

Frank Farance, Farance Inc.
+1 212 486 4700
frank@farance.com

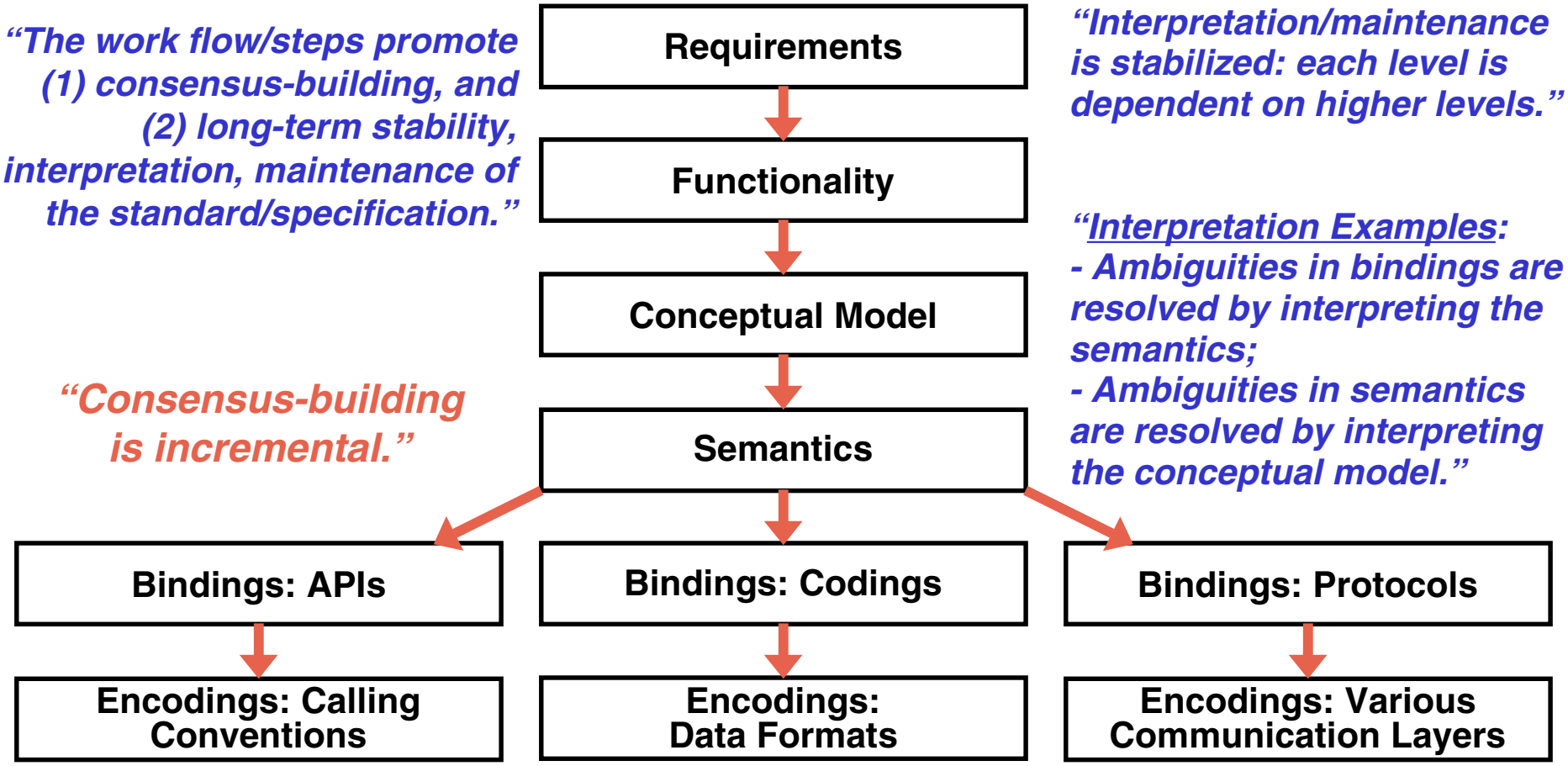
Data Interchange: Methodology

Methodology: Work Flow And Progressive Deliverables

The Steps of Building Successful Information Technology Standards/Specifications

"The work flow/steps promote (1) consensus-building, and (2) long-term stability, interpretation, maintenance of the standard/specification."

"Consensus-building is incremental."



"Interpretation/maintenance is stabilized: each level is dependent on higher levels."

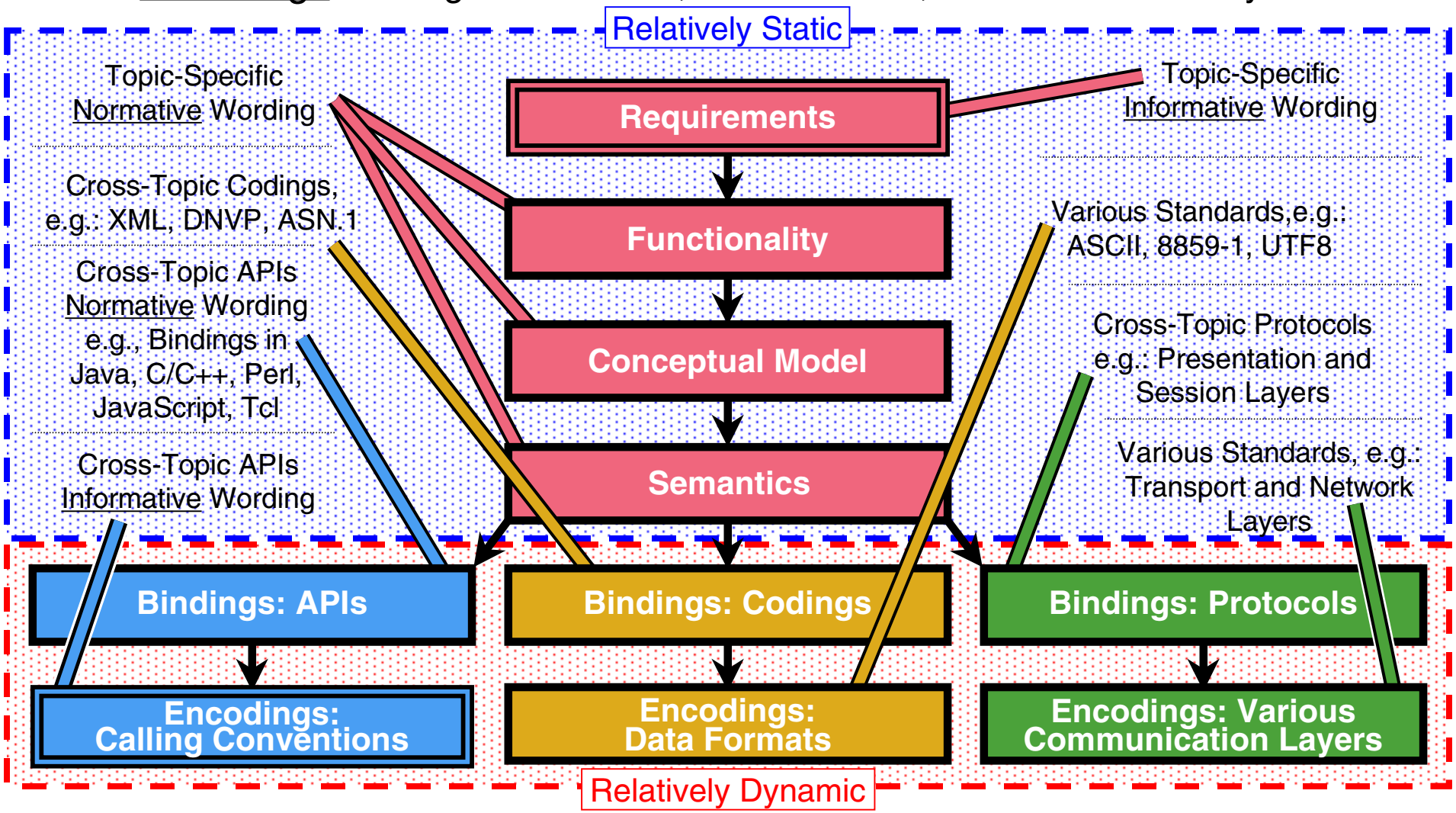
"Interpretation Examples:
- Ambiguities in bindings are resolved by interpreting the semantics;
- Ambiguities in semantics are resolved by interpreting the conceptual model."

A Framework for Harmonized/Consistent ...

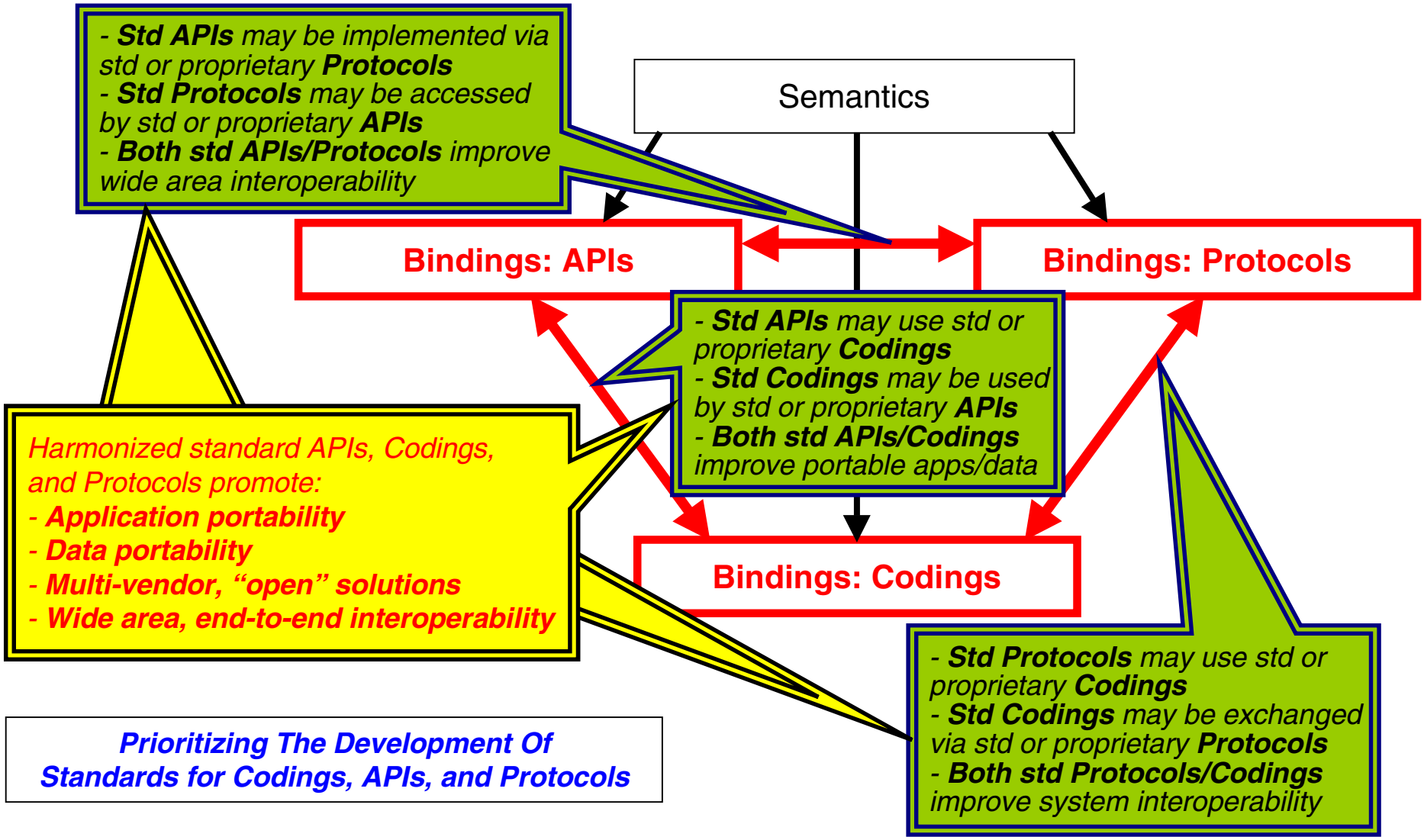
Adaptability: Long Term vs. Short Term

Bindings: APIs, Codings, Protocols

Encodings: Calling Conventions, Data Formats, Communication Layers

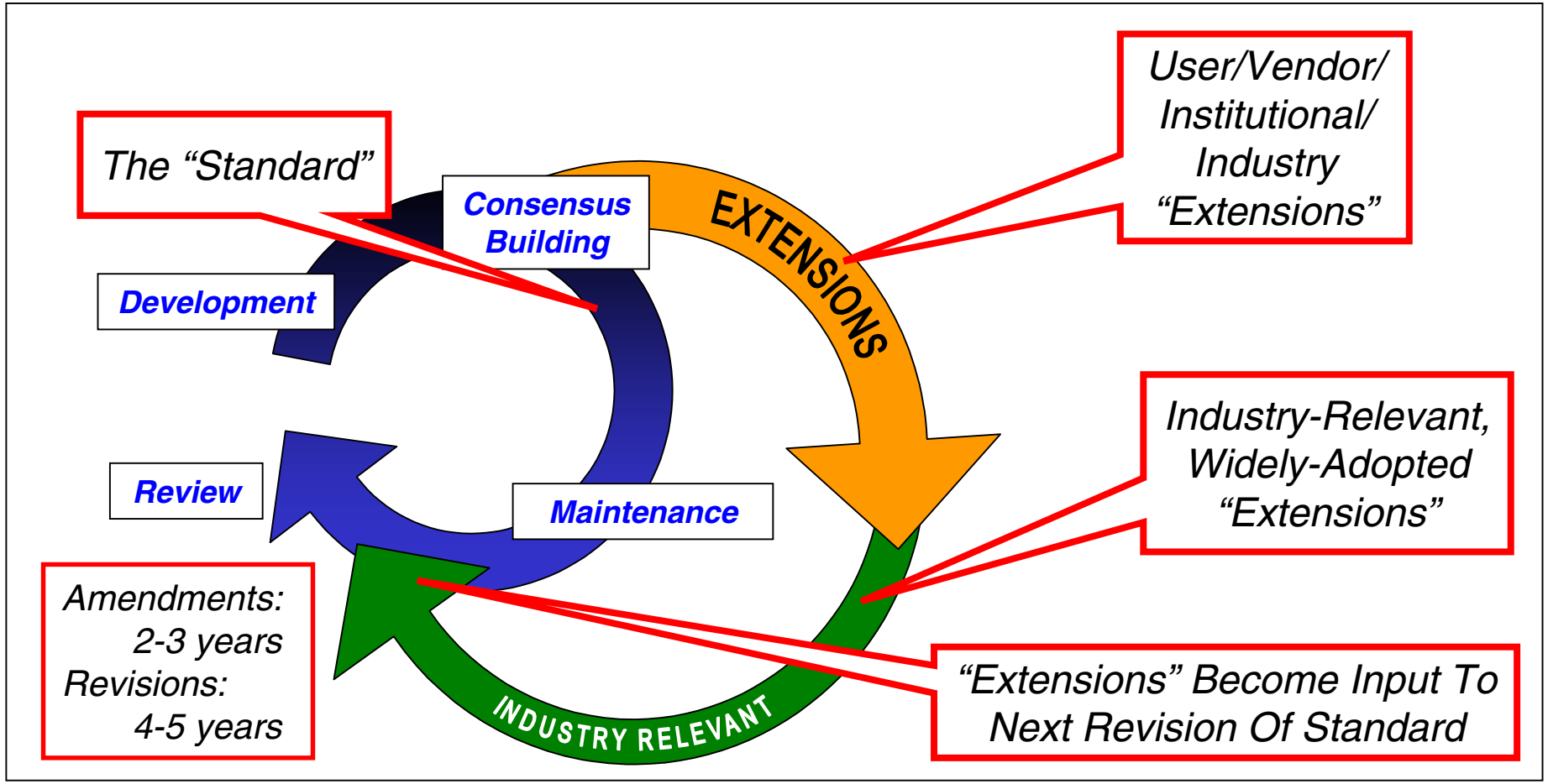


APIs, Codings, Protocols — All Three Are Required

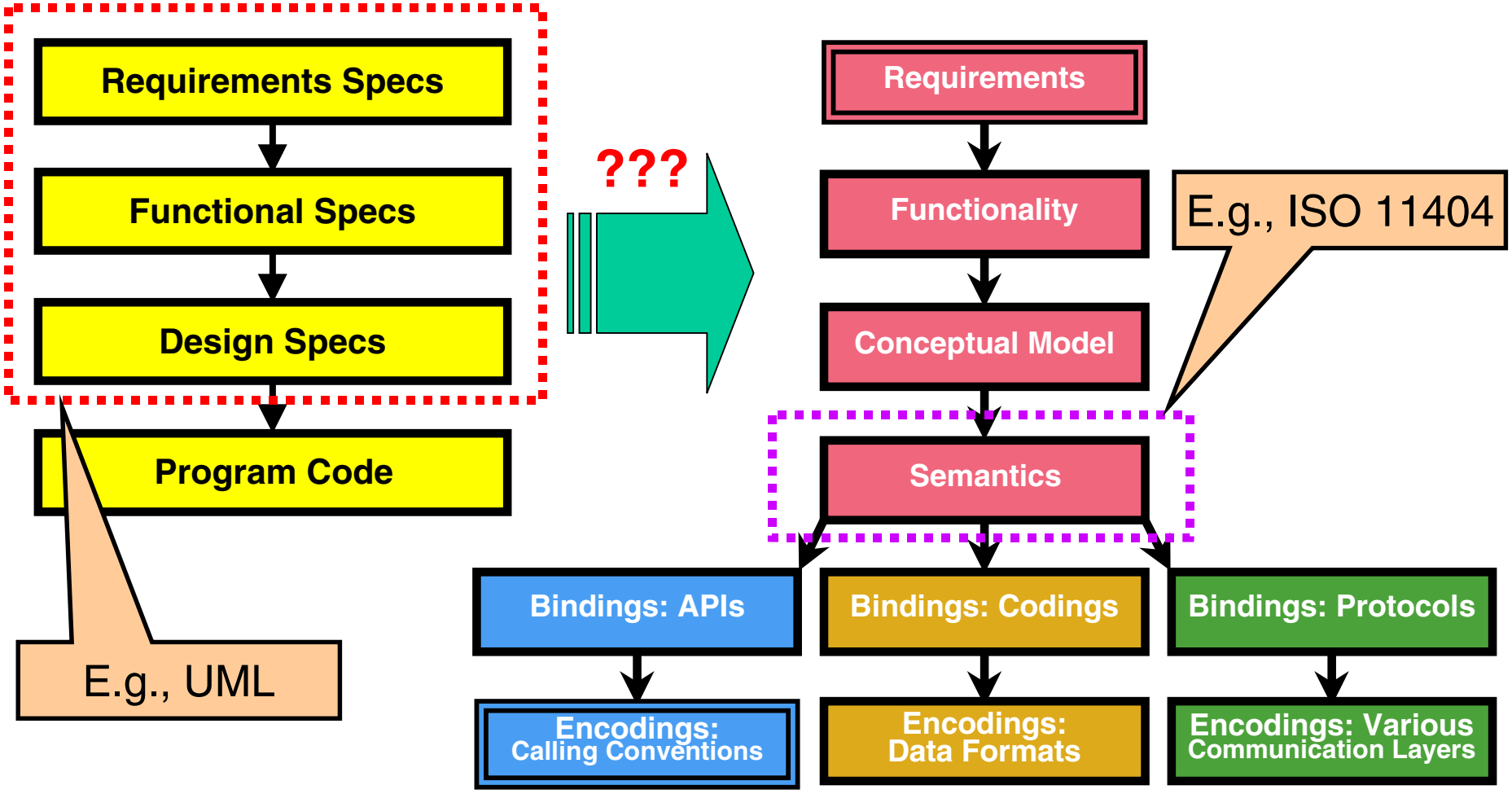


Prioritizing The Development Of Standards for Codings, APIs, and Protocols

Building Standards/Specs In Several Steps



Note: Describing Implementations Is *Different* Than Describing Stds/Specs



Metadata Interchange: Methodology/Interoperability

ISO/IEC 20944-xx
Metadata Interoperability

- Parts:

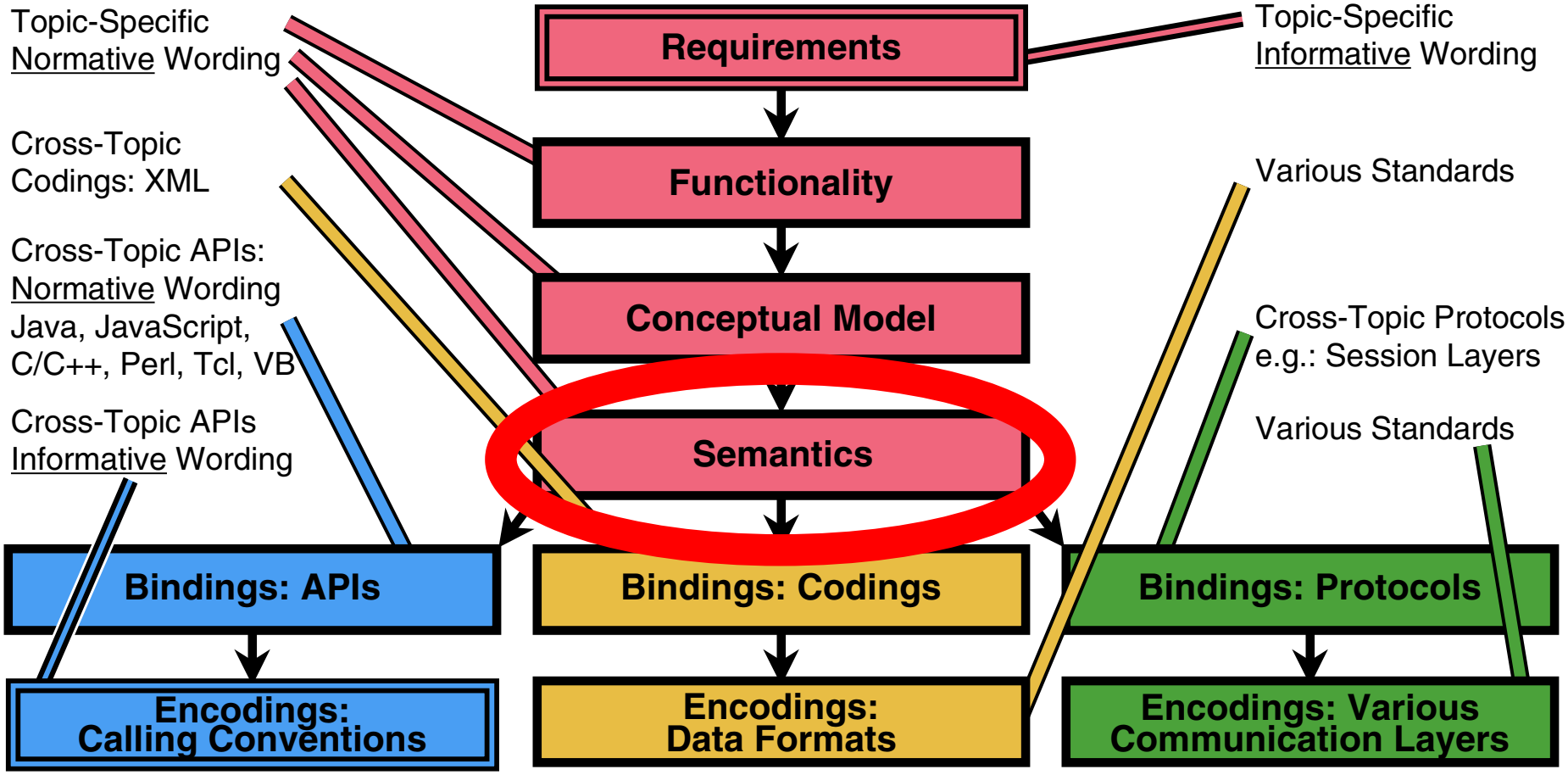
- 00: Overview
- 01: Conformance
- 02: Attribute Mapping

- 20: Coding binding overview
- 21: XML coding binding
- 22: DNVP coding binding
- 23: ASN.1 coding binding
- 24: DCMI coding binding

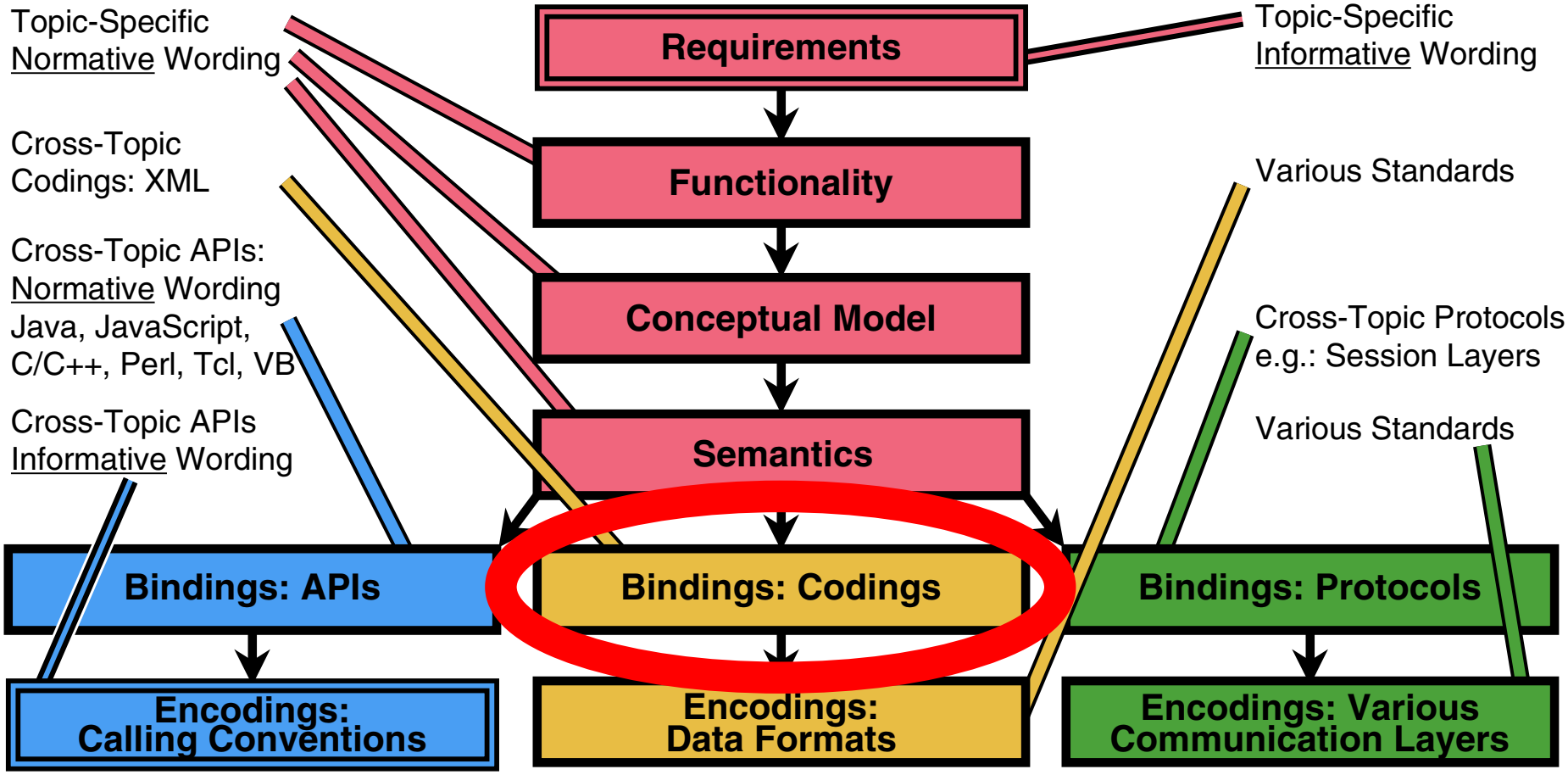
- 40: API binding overview
- 41: C API binding
- 42: C++ API binding
- 43: Java API binding
- 44: JavaScript API binding
- 45: Perl API binding
- 46: LISP API binding

- 60: Protocol binding overview
- 61: ODBC protocol binding
- 62: DCTP protocol binding
- 63: WSDL protocol binding
- 64: LDAP protocol binding

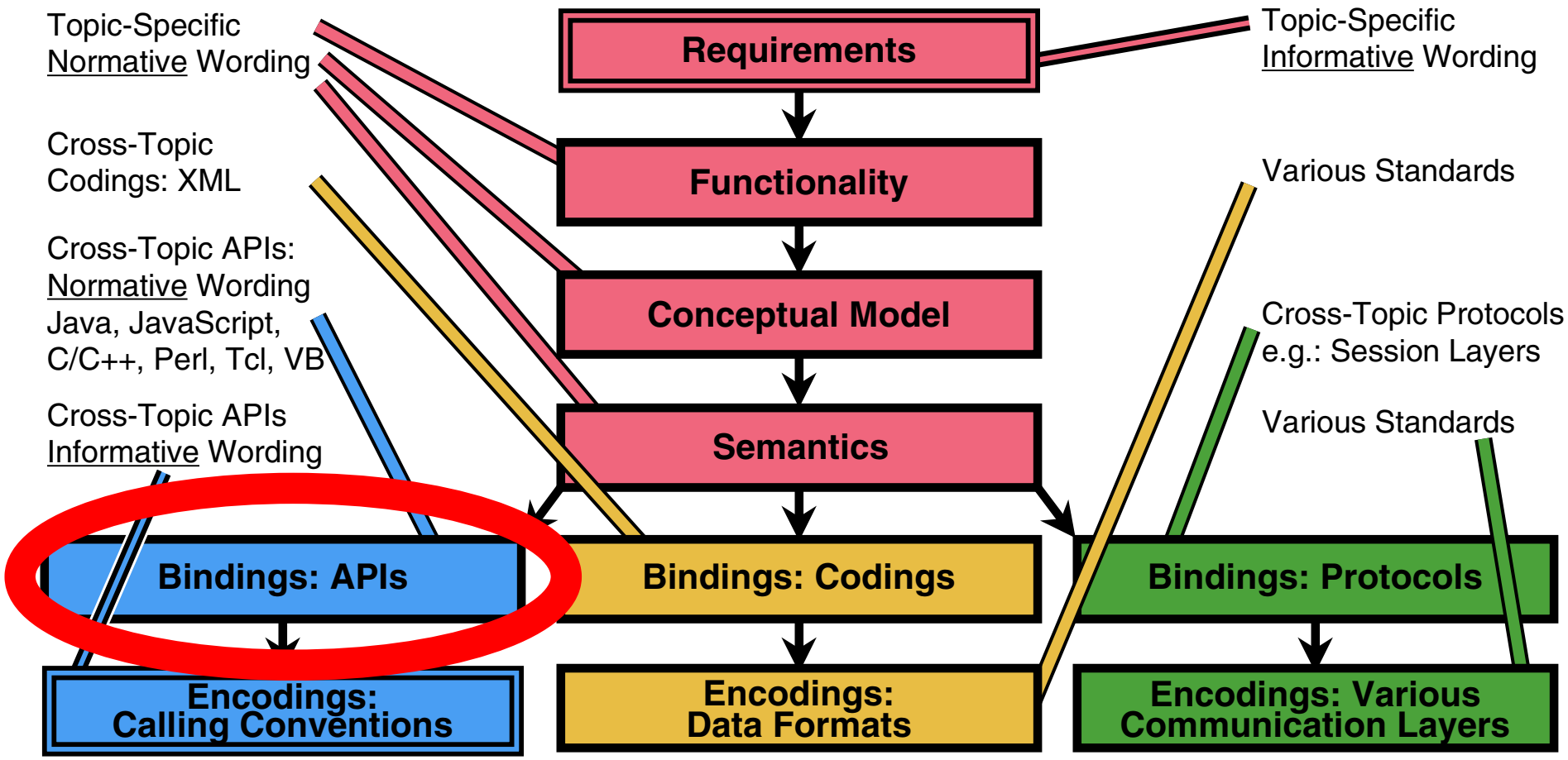
20944-01: Conformance
20944-02: Attribute Mapping



20944-2x Coding Bindings e.g., XML, ASN.1, DNVP



20944-4x API Bindings
e.g., C, Java, Perl



Metadata Access Service (MDAS, ISO/IEC 20944-4x)

- Requirements
  - Make inquiries into repositories to determine metadata
  - Use metadata for further interoperability of repositories
  - Help facilitate metadata/data interchange among repositories
  - Supports semi-structure data access
  - Can support lexicon query service, terminology services

Metadata Access Service (MDAS, ISO/IEC 20944-4x)

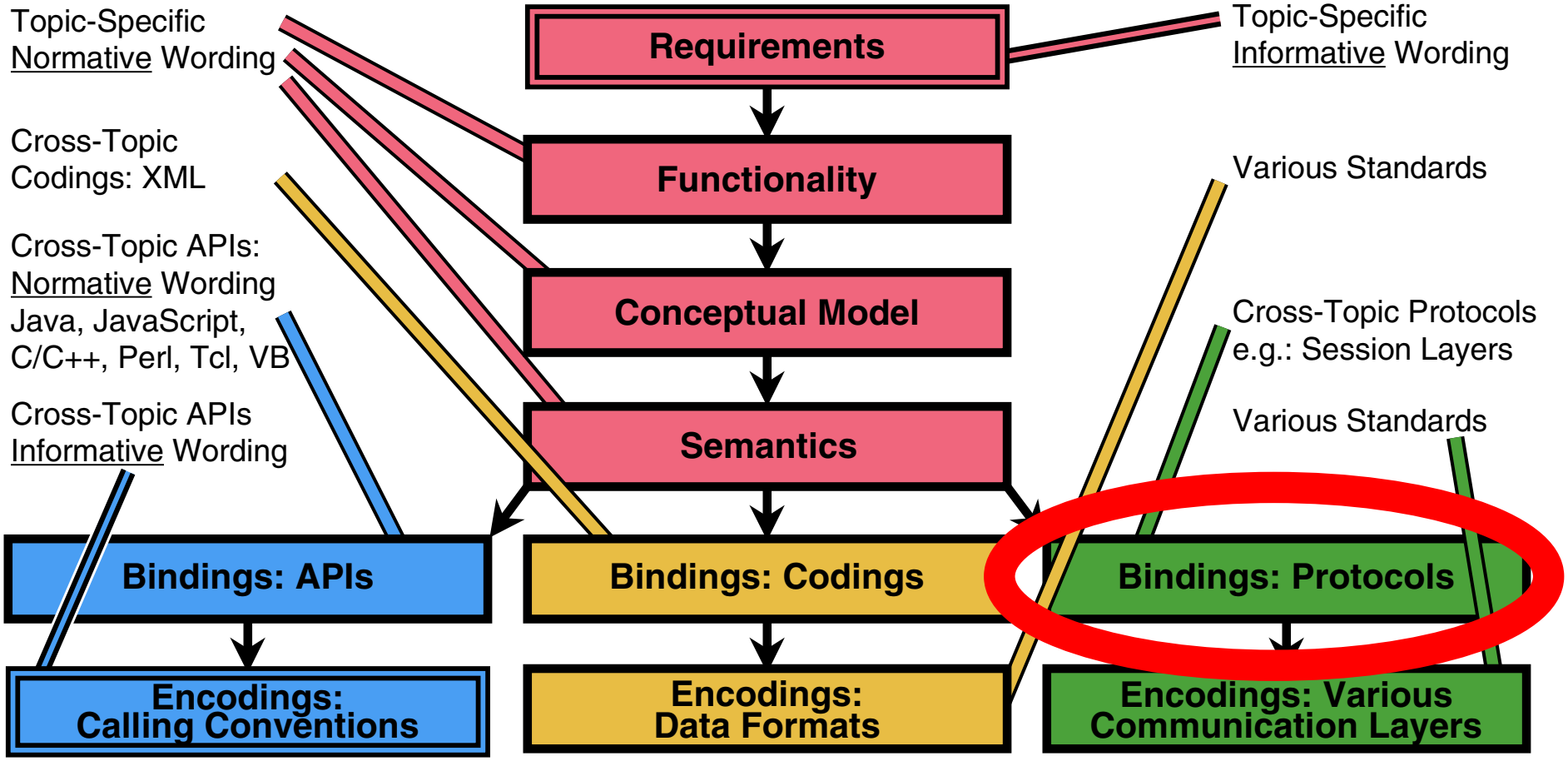
- Functionality
  - Interacts directly with repositories
  - Get (and put) metadata/data
  - Specialized query features to handle:
    - Search by type
    - Search by identifier
    - Search by label
    - Search by property (attribute)

Metadata Access Service (MDAS, ISO/IEC 20944-4x)

- Semantics Summary
  - Currently being refined, based on SDA API, LQS, DCTP, etc.
  - Work being harmonized with ISO 15067-1 (DCTP being incorporated)

- **`CONNECT`:** connect to repository
- **`OPEN`:** begin access to repository
- **`SET`:** set protocol parameters
- **`QUERY`:** query protocol parameters
- **`GIVEAUTH`, `NEEDAUTH`:** authentication
- **`NOMAD`:** nomadic (disconnected) access
- **`PUTPATH`:** change view (directory)
- **`GETVAL`:** get info from repository
- **`PUTVAL`:** put info to repository
- **`LIST`:** retrieve names in repository
- **`EVENT`:** client and server event processing
- **`CLOSE`:** end access to repository
- **`DISCONNECT`:** disconnect from repository

20944-6x Protocol Bindings e.g., ODBC, DCTP, LDAP



Role Of ISO/IEC 11179 For Data Interchange

- Description of data elements in a repository
- Used in actual implementations
- Used in metadata exchange among repositories

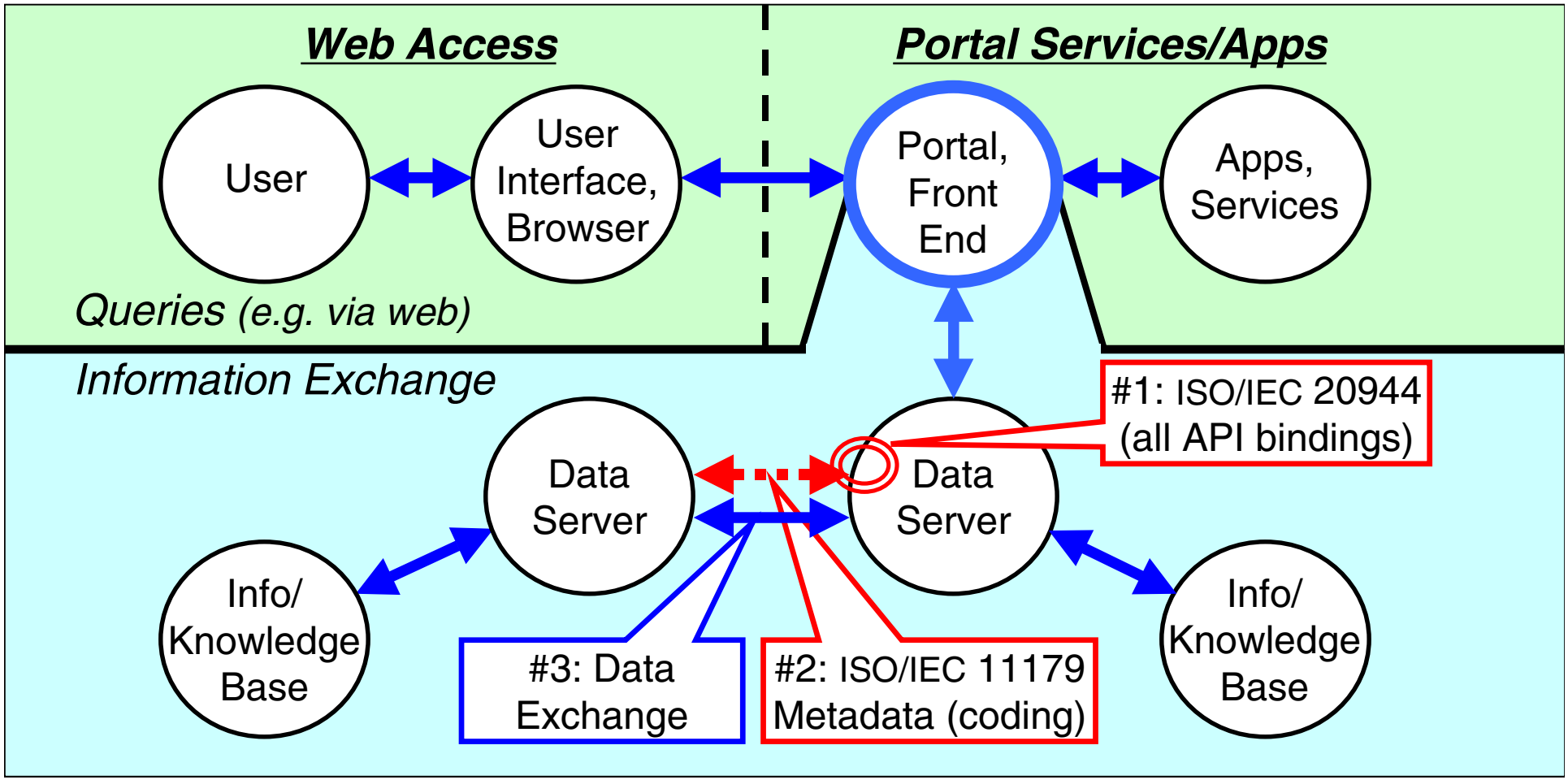
Conceptual Model (High Level)

- #1: Connect to repository; query metadata/data of repository
- #2: Determine what data to access (via metadata) and how to access
- #3: Data exchange

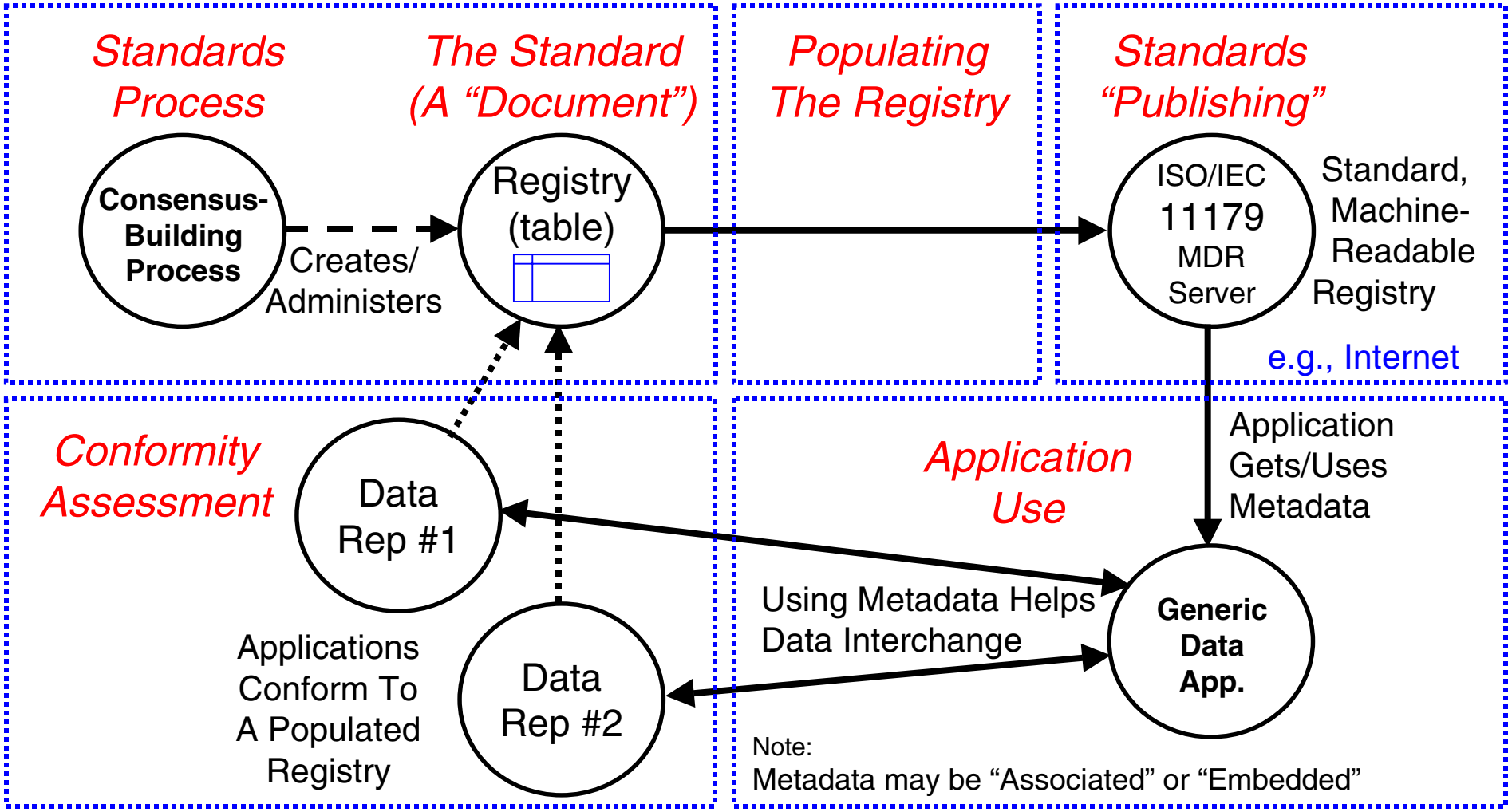
Conceptual Model (Low Level)

- Connect to repository
- Negotiate parameters (security, formats, etc.)
- Navigate the repository
- Get (and put) information
- Merge data/metadata/props namespace
- Miscellaneous data mgmt. operations

Example Of 20944 API, Then ISO/IEC 11179 Metadata Exchange, Then Data Exchange

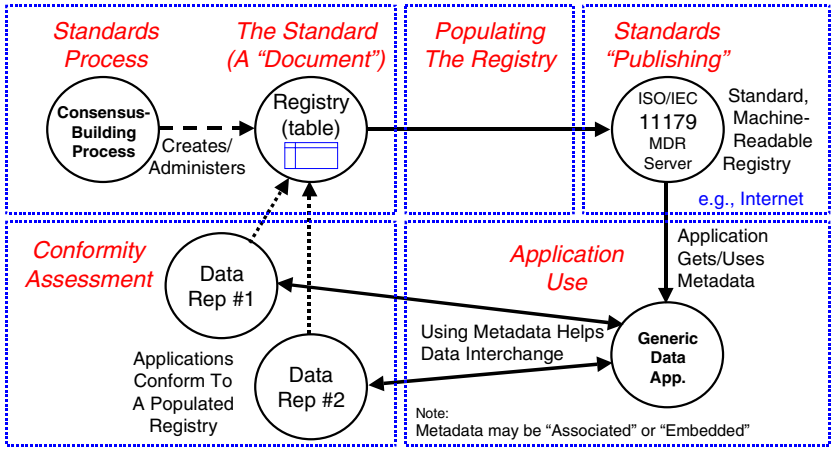


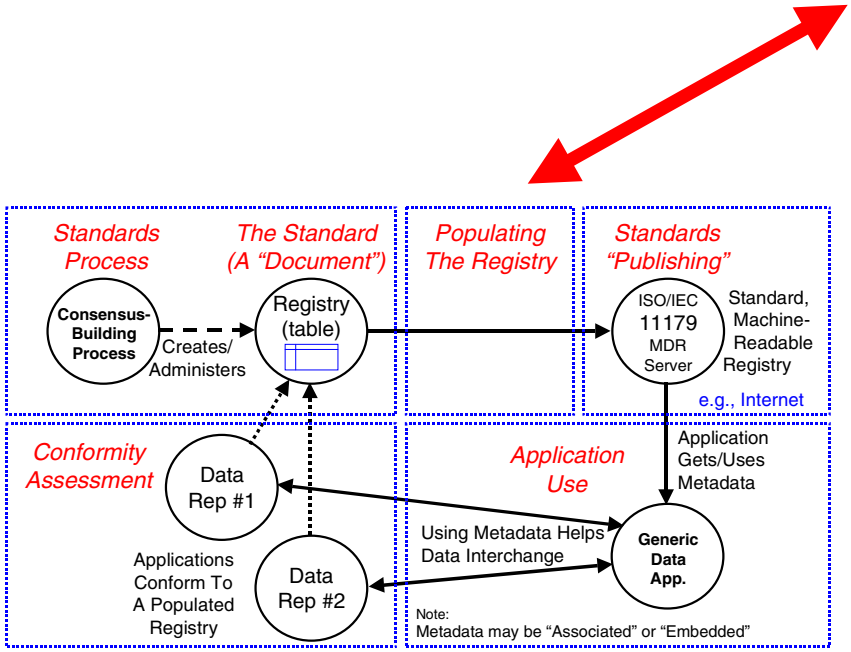
Data Model, Registration, Metadata Registries, and Interoperability



Using 11179 for Namespaces And Other Administered Items

ISO/IEC 11179 Used For Namespaces





ISO/IEC 11179 Used For Other Administered Items, e.g., XML tags, data elements, etc..

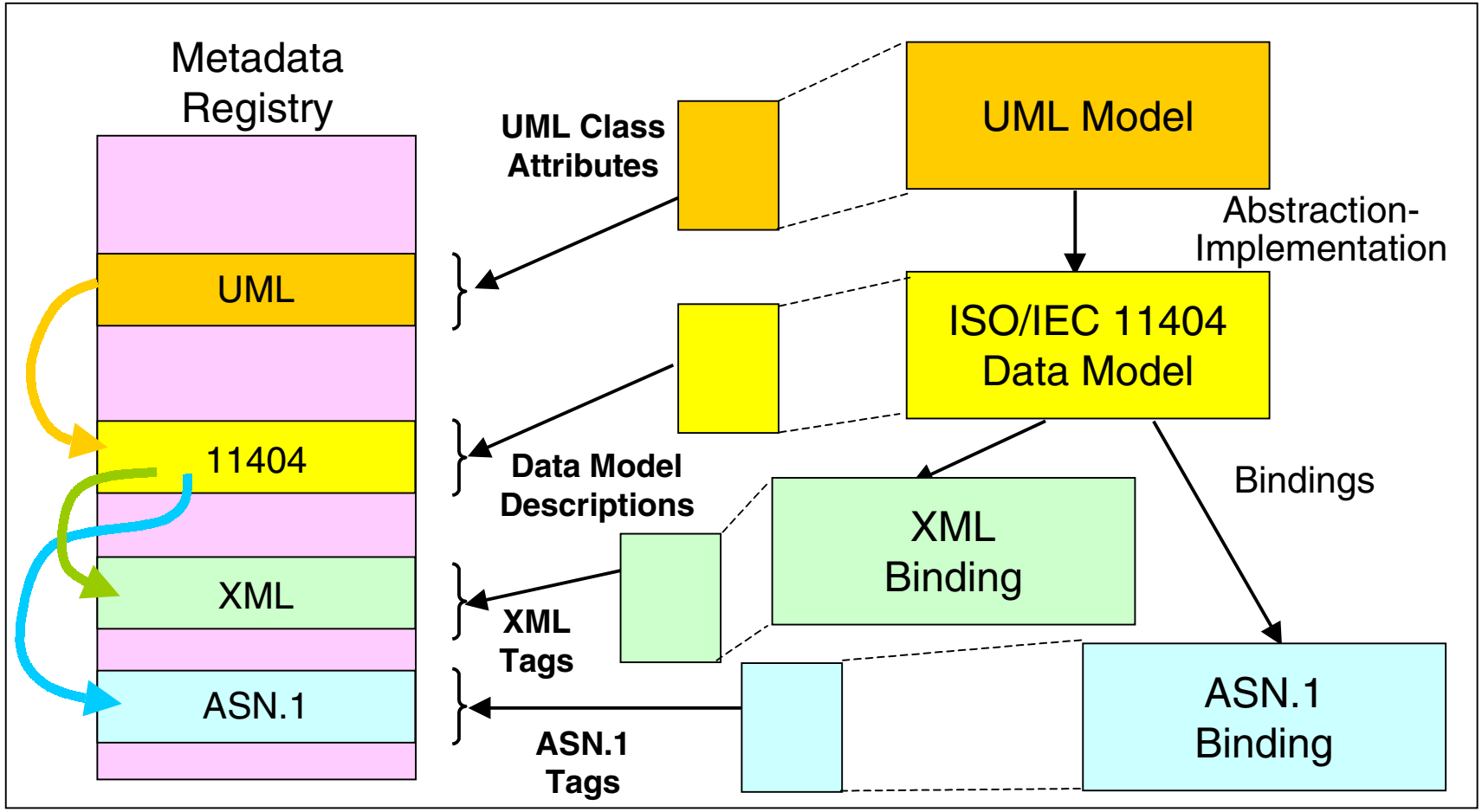
Sample MDR Access

```
handle = mdas_connect("repository");
session = mdas_open(handle, "read");
list = mdas_list(session, "registry_1");
mdas_put_path(session, "registry_1");
for ( i = 0 ; i < list.length() ; i++ )
{
        adm_item = list[i];
        adm_session = mdas_open(adm_item, "read");
        mdas_put_path(adm_session, adm_item);
        dump_metadata(adm_session, adm_item);
        mdas_close(adm_session);
}
mdas_close(session);
mdas_disconnect(handle);
```

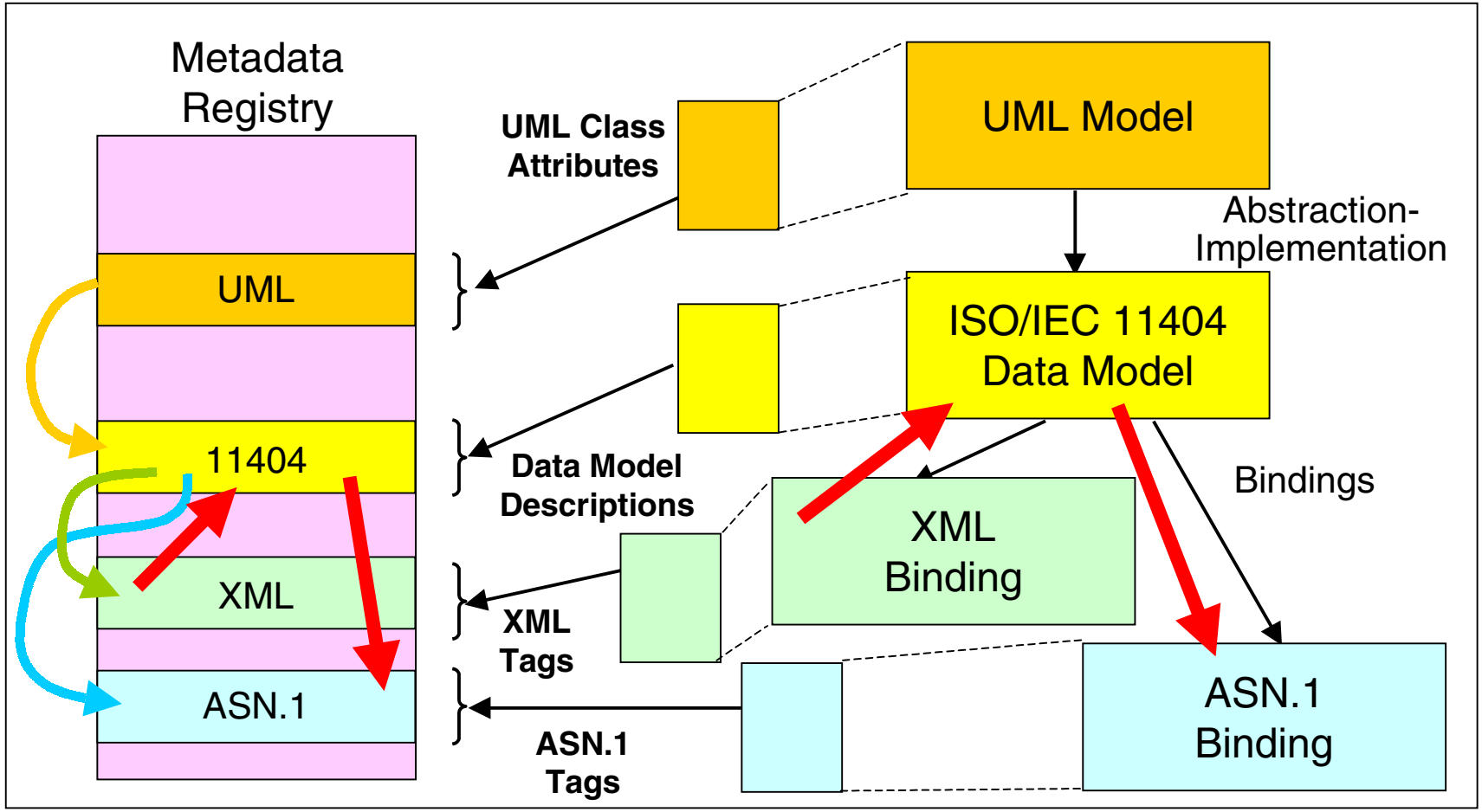
Sample MDR Access

```
dump_metadata(adm_session, adm_item)
{
        printf("%s, registration_status: %s\n", adm_item,
                mdas_get_value_as_string("registration_status");
        printf("%s, item_identifier.data_identifier: %s\n", adm_item,
                mdas_get_value_as_string("item_identifier/data_identifier");
        printf("%s, item_identifier.version: %s\n", adm_item,
                mdas_get_value_as_string("item_identifier/version");
        printf("%s, submitted_by.organization.name: %s\n", adm_item,
                mdas_get_value_as_string("submitted_by/organization/name");
        printf("%s, submitted_by.organization.address: %s\n", adm_item,
                mdas_get_value_as_string("submitted_by/organization/address");
}
```

Relationship Among Metadata



Example: Automated Translation — Made Possible By Methodology/Interoperability



Namespaces: Theory

Main Features of Namespaces

- Much research in past ~40 years
- Main features of “namespaces”
  - Naming conventions (periods vs. slashes)
  - Context association (implicit rules)
  - Context specification (paths)
  - Scope (visibility/invisibility)
  - Resolution (transform name to particular reference)
  - Dereference (semantics on use of name)

Namespace Background

- What is a name?
  - A designator: e.g., an identifier or symbol
- What is a namespace?
  - A pool of names within a context
  - Must be context-associated (otherwise it would be context-free names; typically, difficult)

Namespace Background

- Why namespace?
  - Namespaces are, typically, a solution to a scaling problem: as the number of entity-designator associations increases, the complexity increases
  - Sometimes, namespaces aren't necessary when other global/centralized naming conventions are possible (e.g.,lat-lon-alt)

Namespace Background

- Each environment has its own incorporation of “context”:
  - e.g., some approaches uses prefixes
  - e.g., some approaches use rule-based systems (e.g., C++ overloading)
- Context specification is important
  - Naming conventions, e.g., “namespace separators” (periods, slashes, etc.)

Namespace Background

- Where can names be used?
  - Scope
  - Conflicts
- Automation
  - How are names transformed to “real stuff”?
  - Most people/projects gloss over this stuff

Namespace Background

- Example: JavaScript Scope Chain and Identifier Resolution
  - Every execution context has associated with it a scope chain. A scope chain is a list of objects that are searched when evaluating an *Identifier*. When control enters an execution context, a scope chain is created and populated with an initial set of objects, depending on the type of code. During execution within an execution context, the scope chain of the execution context is affected only by **`with`** statements and catch clauses.

Namespace Background

- During execution, the syntactic production *PrimaryExpression : Identifier* is evaluated using the following algorithm:
  - 1. Get the next object in the scope chain. If there isn't one, go to step 5.
  - 2. Call the [[HasProperty]] method of Result(1), passing the Identifier as the property.
  - 3. If Result(2) is true, return a value of type Reference whose base object is Result(1) and whose property name is the Identifier.
  - 4. Go to step 1.
  - 5. Return a value of type Reference whose base object is null and whose property name is the Identifier.
- The result of evaluating an identifier is always a value of type Reference with its member name component equal to the identifier string.

Evaluation Criteria for Using/Documenting Namespaces?

- What are the naming conventions?
- How is context associated?
- How is context specified/described?
- What are the scoping rules?
- What is the *algorithm* for resolving names?
- How are names dereferenced?
- What does name dereferencing mean?

What Works

- Most programming languages do this well
  - Why? Requires automated processing to work
- Internet domain naming
  - Distributed, multi-tiered — works well
- XML namespaces
  - Works poorly, poorly defined, no automation
  - Why did XML namespaces “get it wrong”, yet ~40 years of established practice? How can one “get it right”?

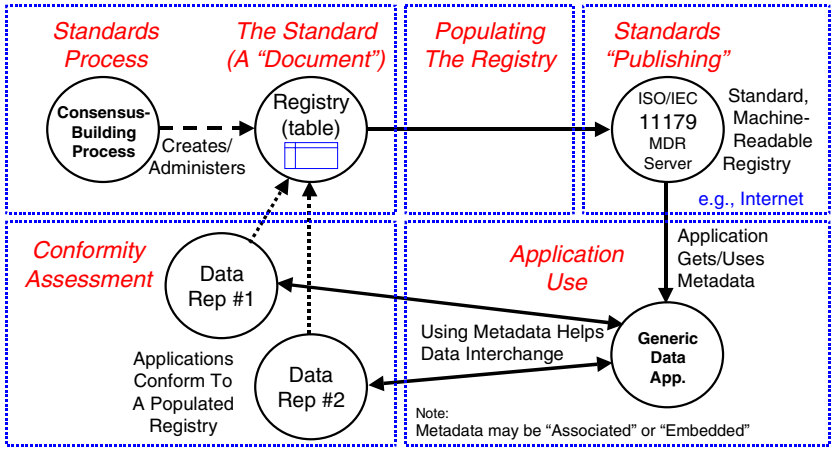
Namespaces: Recommendations

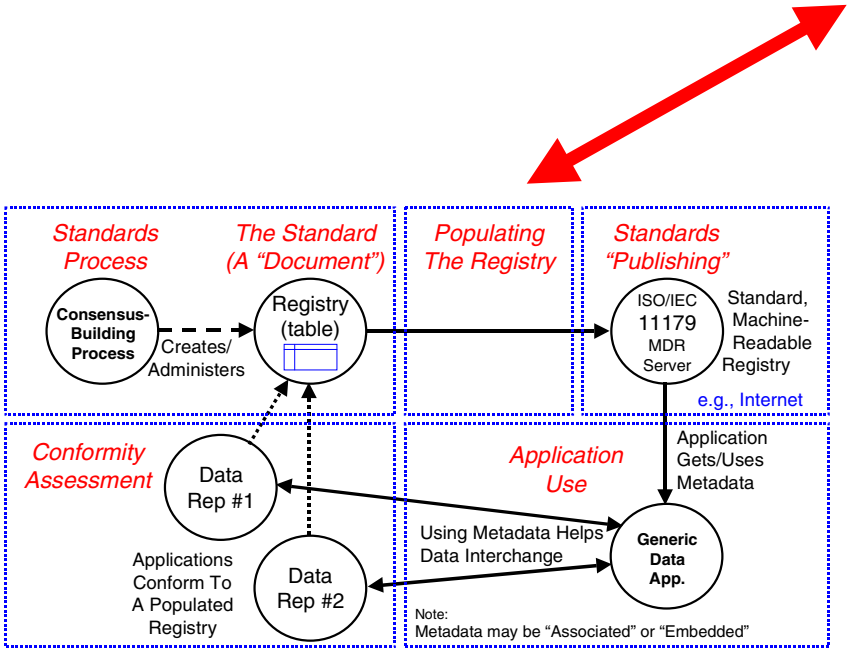
XML Tagging

- Use 11179 for registration of XML tags, schema, etc.
- Use linking among abstraction layers (architecture, design, bindings)
- Use namespaces (in a general way) as a 11179 context for administered items
- Register namespaces as another tier of “data that describes data” (metadata)
- Implementation-specific name resolution

Recap: Using 11179 for Namespaces And Other Administered Items

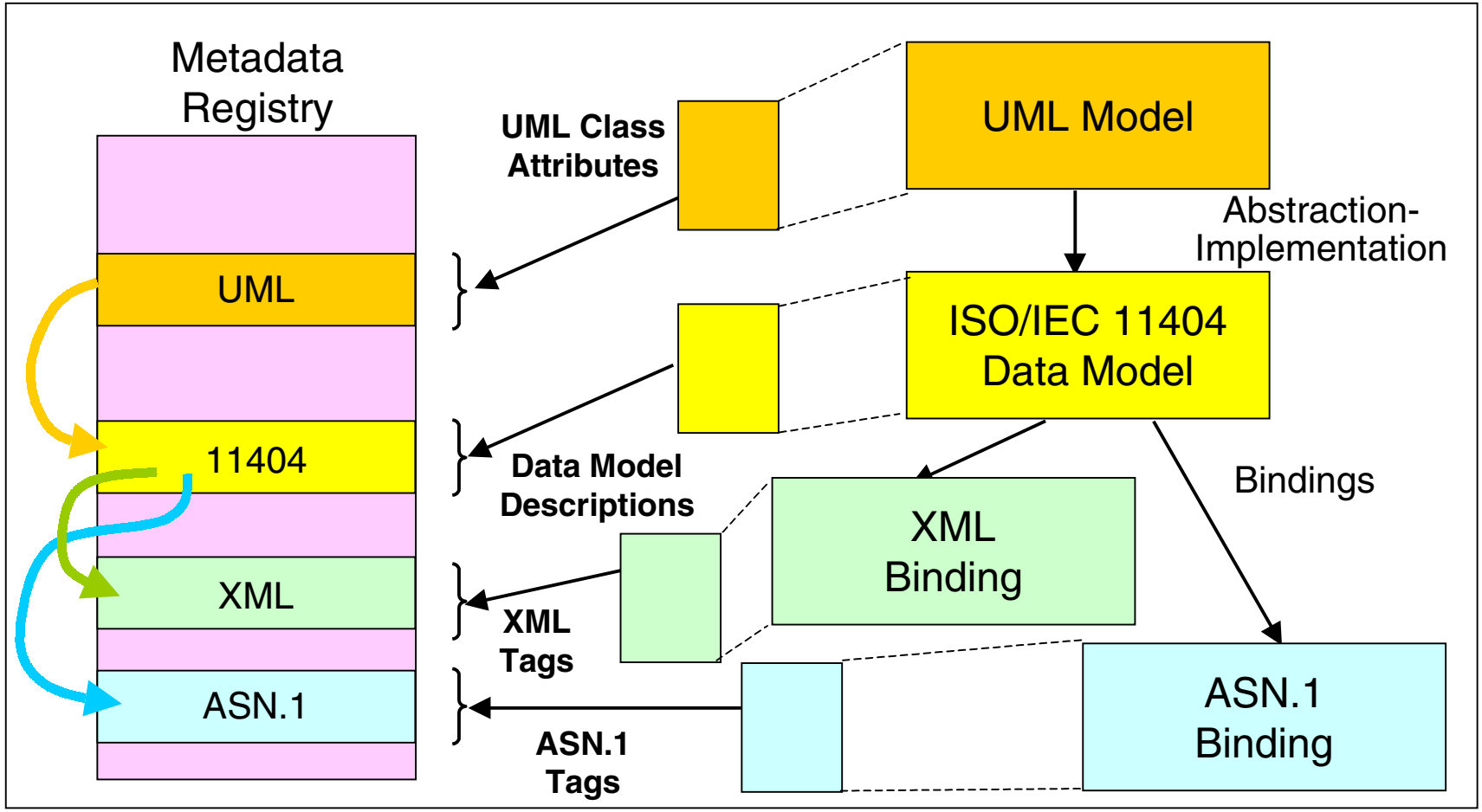
ISO/IEC 11179 Used For Namespaces





ISO/IEC 11179 Used For Other Administered Items, e.g., XML tags, data elements, etc..

Recap: Relationship Among Metadata



Incorporating Namespaces

- Use the evaluation/documentation criteria
  - What are the naming conventions?
  - How is context associated?
  - How is context specified/described?
  - What are the scoping rules?
  - What is the *algorithm* for resolving names?
  - How are names dereferenced?
  - What does name dereferencing mean?

Using XML Namespaces

- Use with care
- Long names work well, too
  - Use uniqueness scheme, e.g., reverse DNS naming mangling or ISO object identifiers

Improvements to 11179-5

- Conformance “object of desire” in 11179-5:
  - A namespace convention
  - Not a particular registry
- Add the namespace evaluation/ documentation criteria to 11179-5

Consensus-Building Processes

- Need consensus-building processes to support registration
- Examples of extremes:
  - international standardization
  - personal registration (squatter)
- Both extremes have strengths/ weaknesses
- Possible: different process for namespace than for (say) data elements

End

- Questions?
- Thank you!