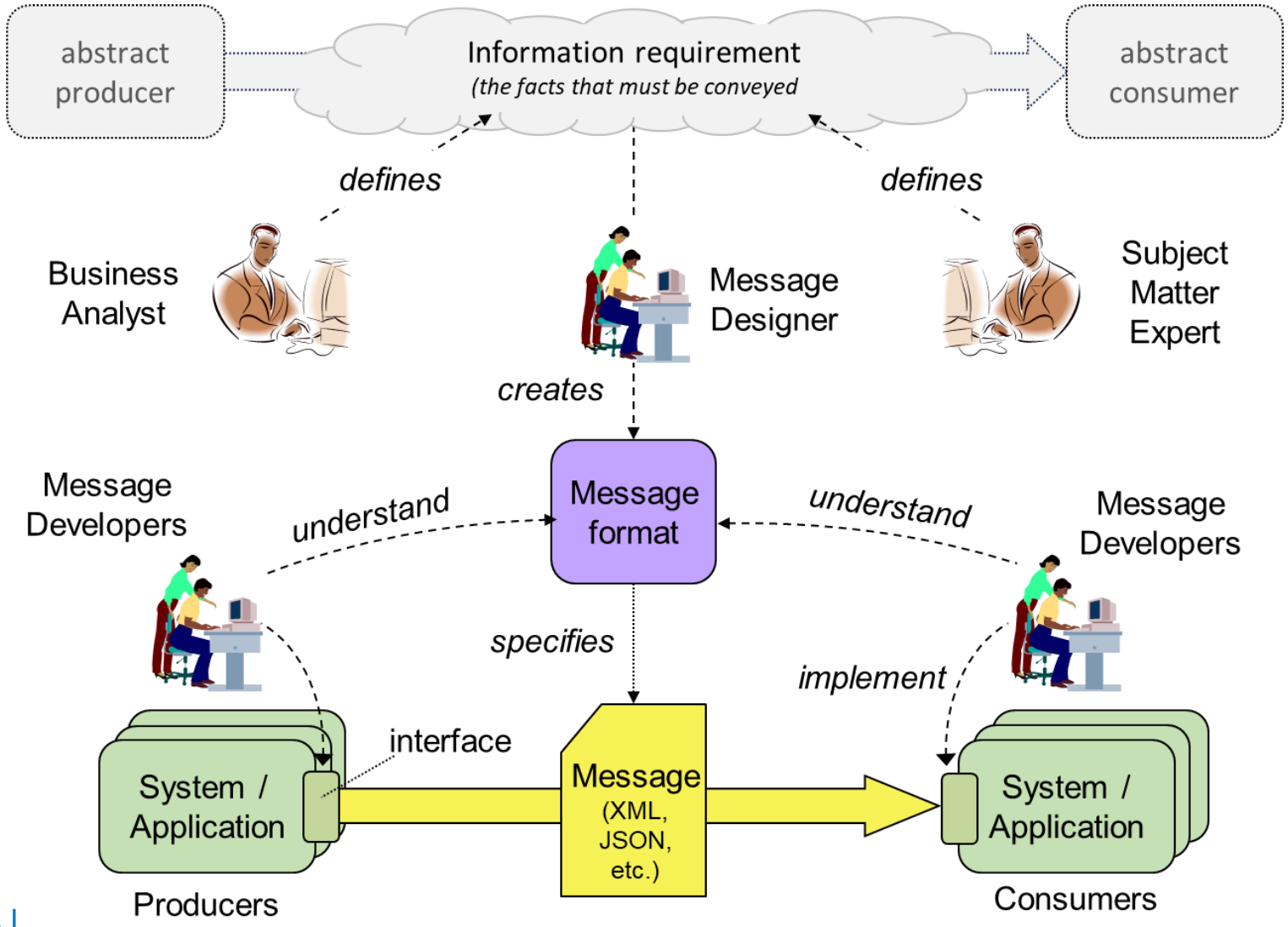
NIEMOPEN

NIEM and Ontologies

Dr. Scott Renner <sar@mitre.org>

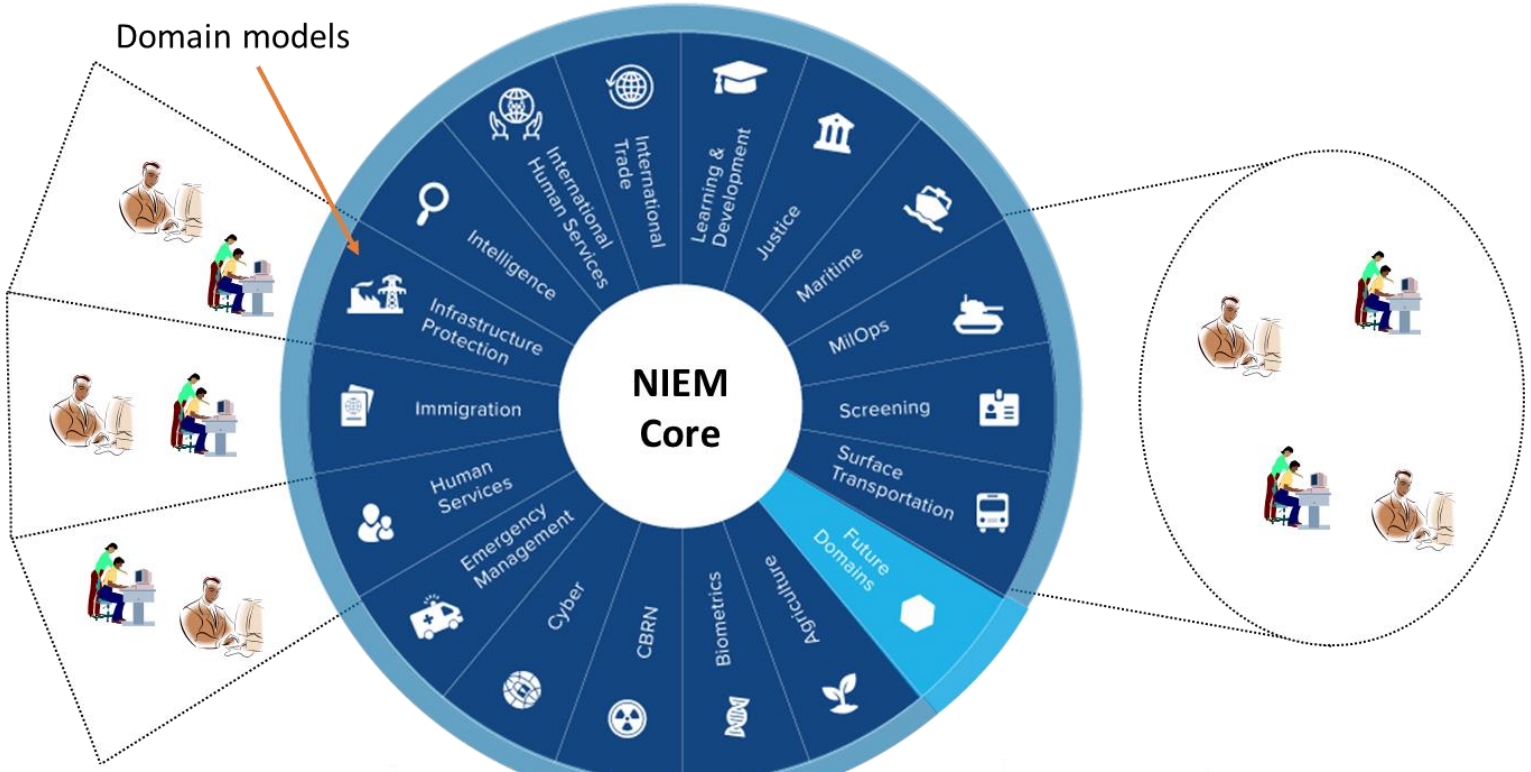
presented at the DIOWG Summit
8 May 2024

NIEM-BASED DATA EXCHANGE



NIEMOPEN

ROLES IN NIEM MODEL DESIGN



Domain communities: analysts, SMEs, developers with shared subject interests

NIEM data model

Business architecture committee (NBAC): representatives from all communities, looking for commonalities

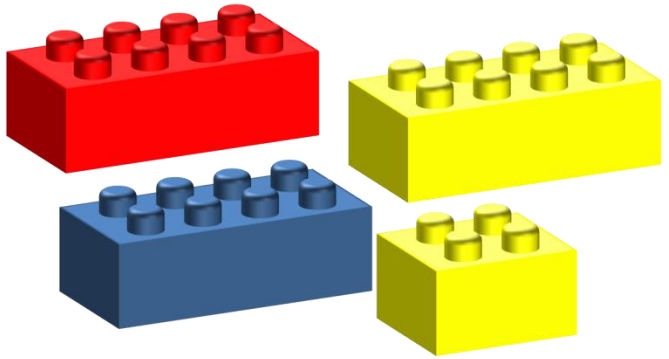
design

design

NIEMOPEN

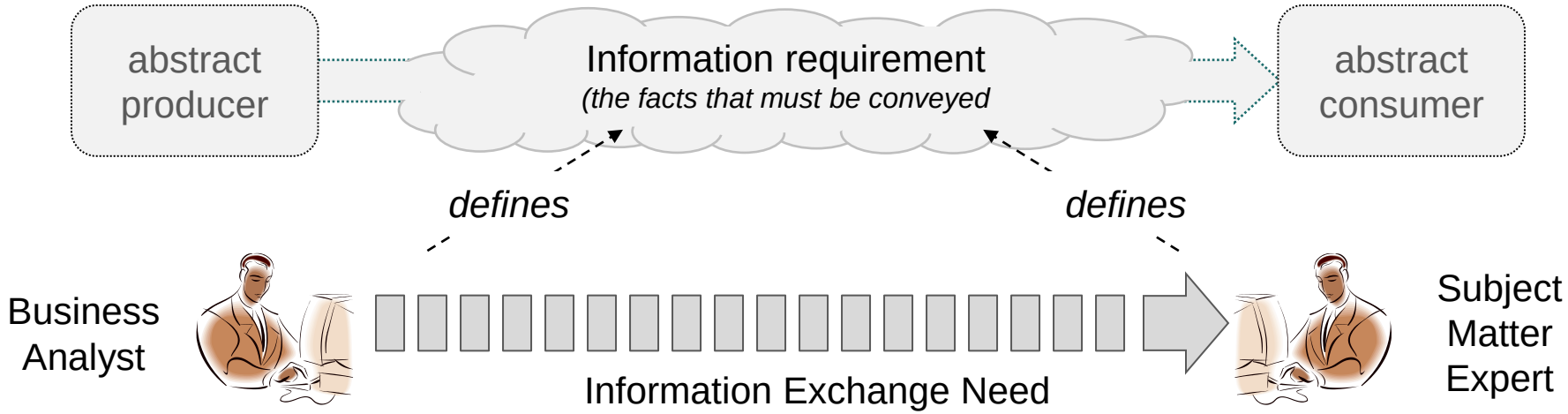
“LEGO™ BLOCK” MODELING

- **Designing an information exchange with NIEM is like building a plastic model out of Lego blocks**
  - You decide what the model should look like
  - You choose the blocks you need for your model
  - Technical specification ensures that the blocks will snap together
- **NIEM is even better than Legos, because**
  - Each NIEM domain provides a collection of useful blocks
  - You can easily create any block that’s missing from your set (and then share it with others)
  - The blocks are all free



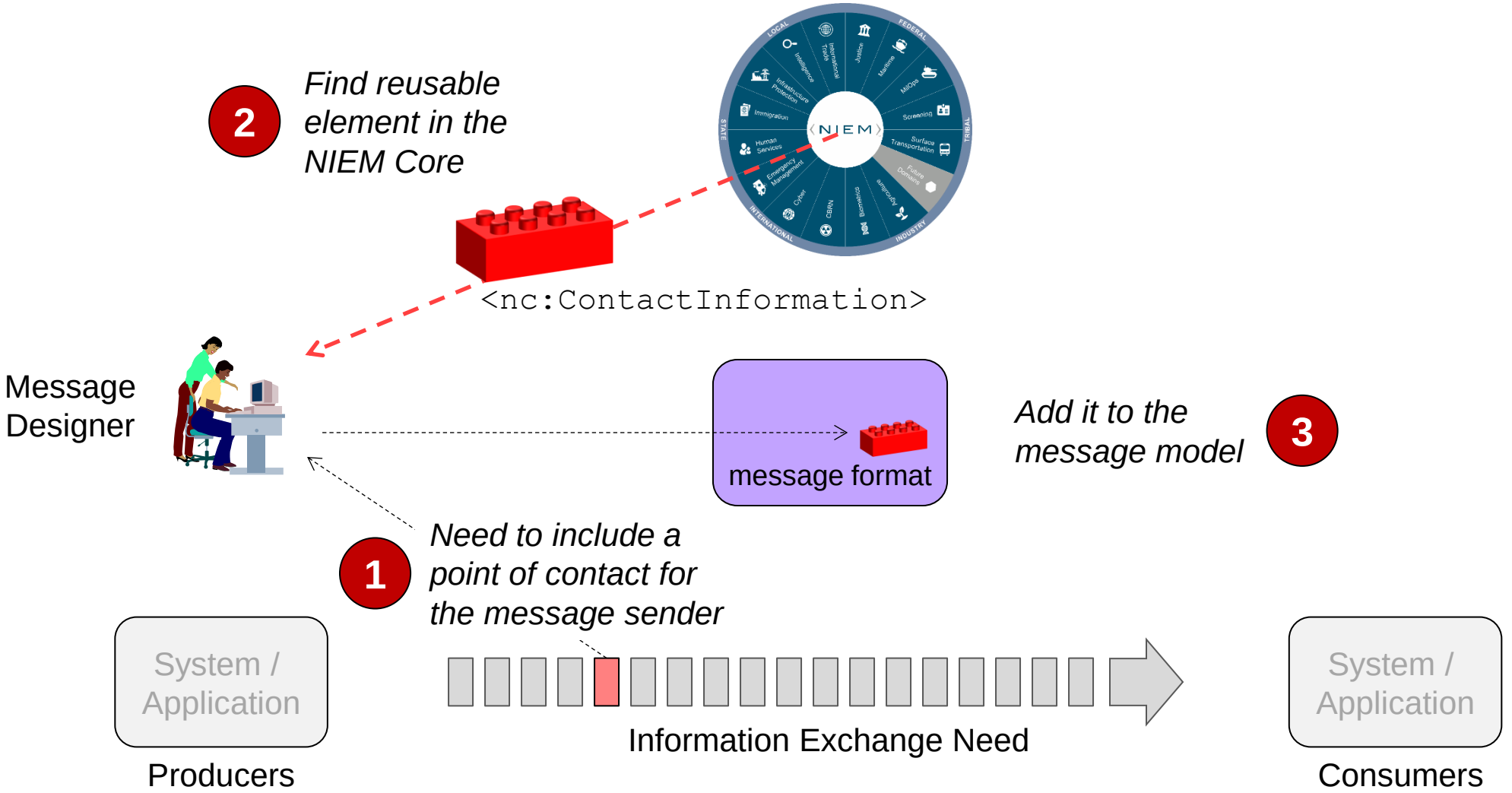
DESIGNING A MESSAGE FORMAT

Start with an information exchange need. You don't have to know all of the participants in advance. It's enough to know that a producer has data that needs to be shared with someone



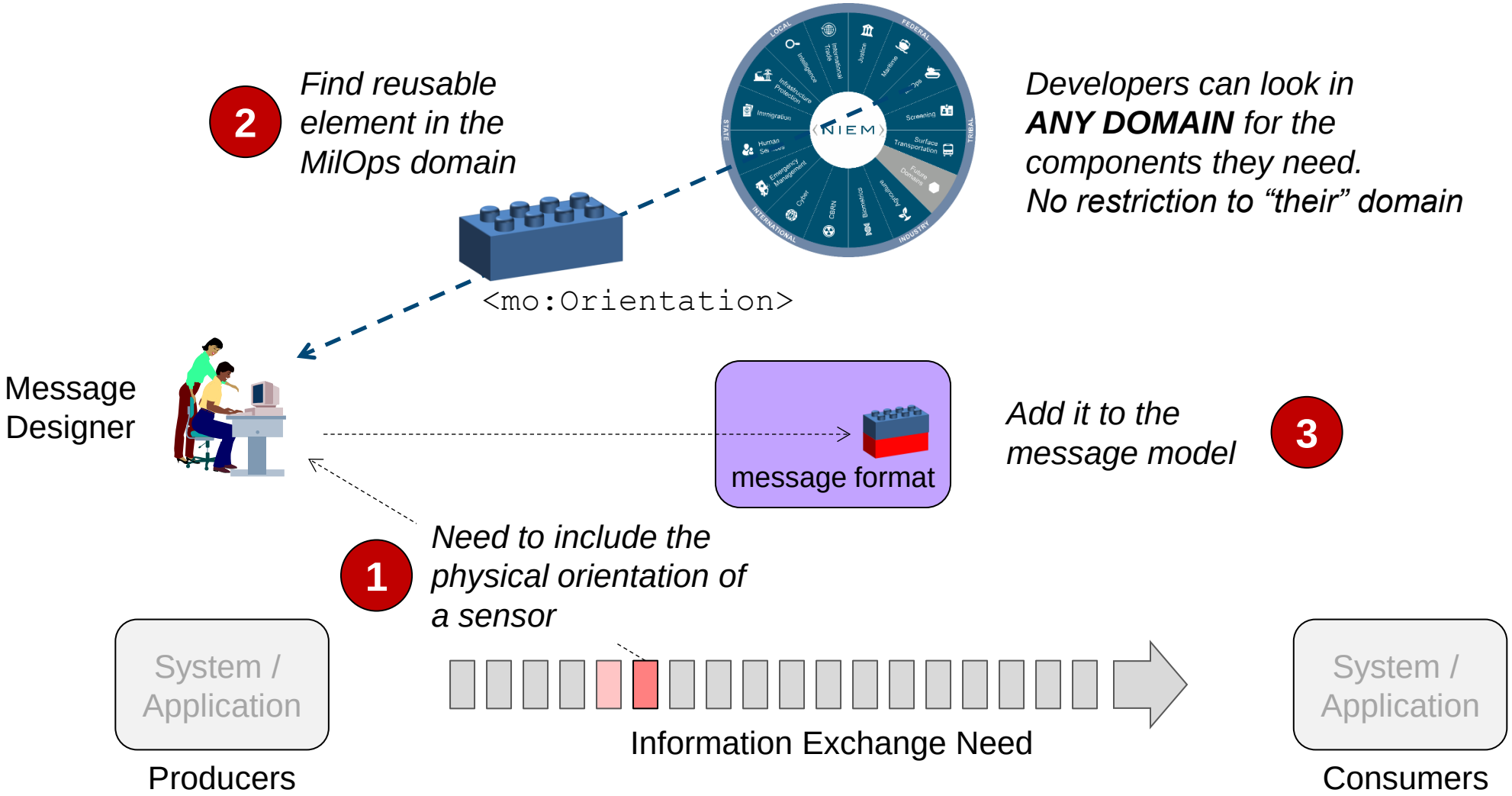
NIEMOPEN

REUSE CORE COMPONENTS FOR DATA NEEDS



NIEMOPEN

REUSE DOMAIN COMPONENTS FOR DATA NEEDS

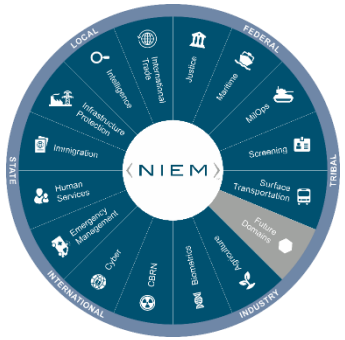


NIEMOPEN

CREATE NEW COMPONENTS FOR DATA NEEDS

2

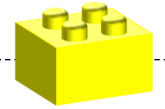
Can't find a matching component in core or domains

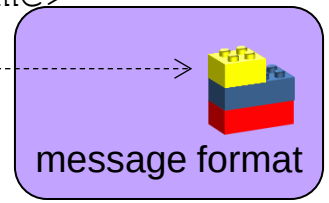


*Developers can create **ANY COMPONENT** they need for their exchange. No restriction to what's already in the NIEM model.*

Message Designer

`<cot:SensorModelName>`





Create and add a new component to the message model

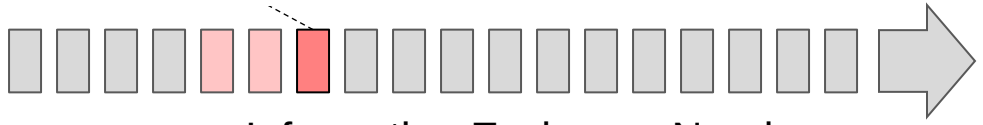
3

1

Need to include the model name of the reporting sensor

System / Application

Producers



Information Exchange Need

System / Application

Consumers

NIEMOPEN

A COMPLETED EXCHANGE SPECIFICATION

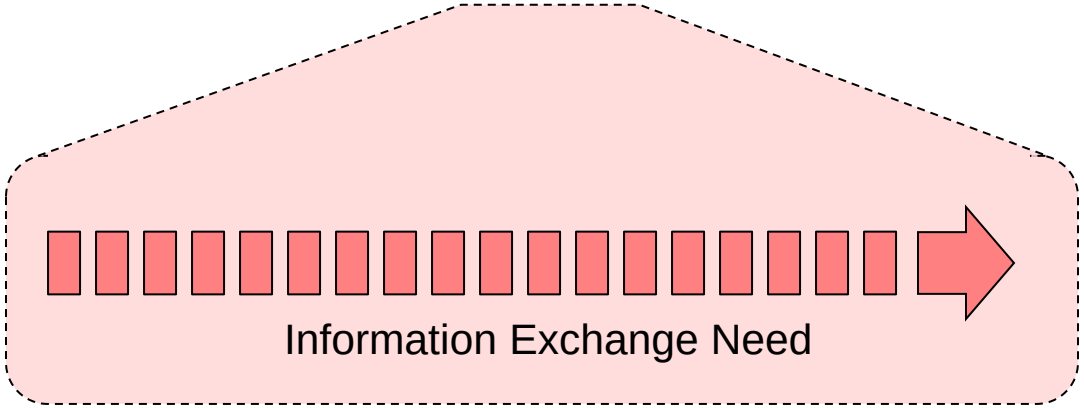
The message format is done when it includes a representation for all aspects of the exchange need

Message Designer

message format

System / Application

Producers



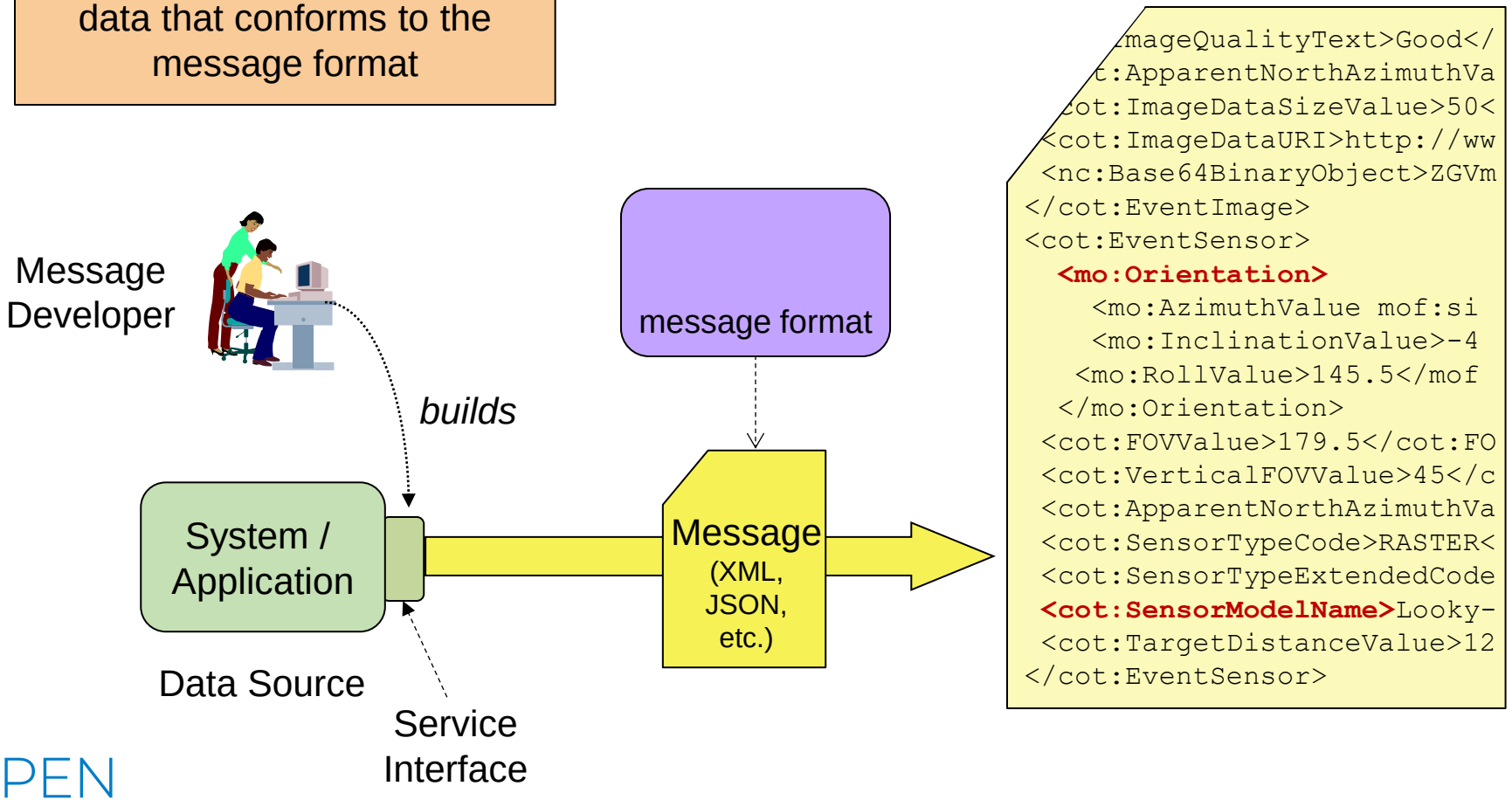
System / Application

Consumers

NIEMOPEN

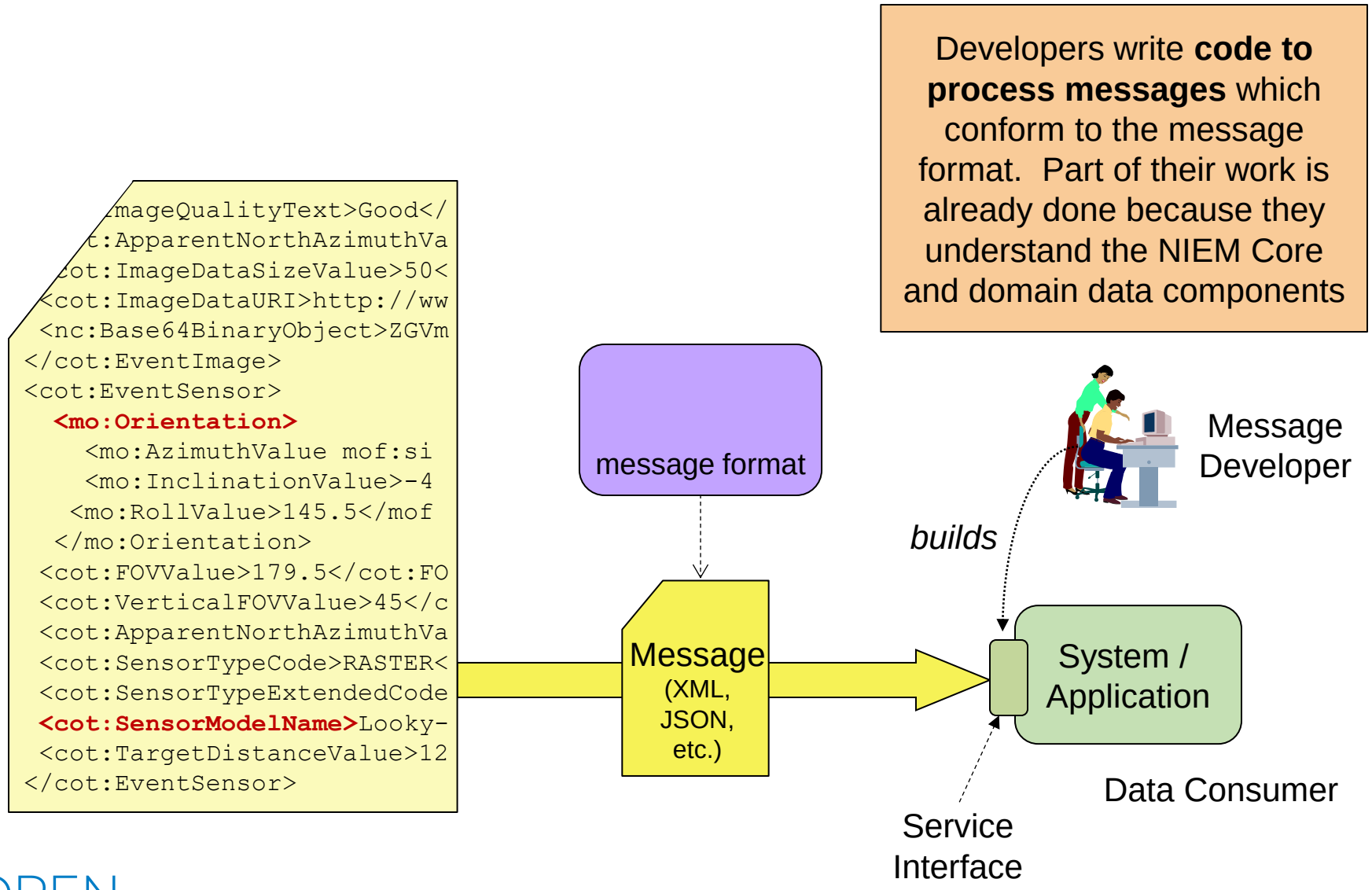
PRODUCER'S SERVICE INTERFACE

Developers write **code to export the source's data**, in the form of XML or JSON data that conforms to the message format



NIEMOPEN

CONSUMER'S SERVICE INTERFACE

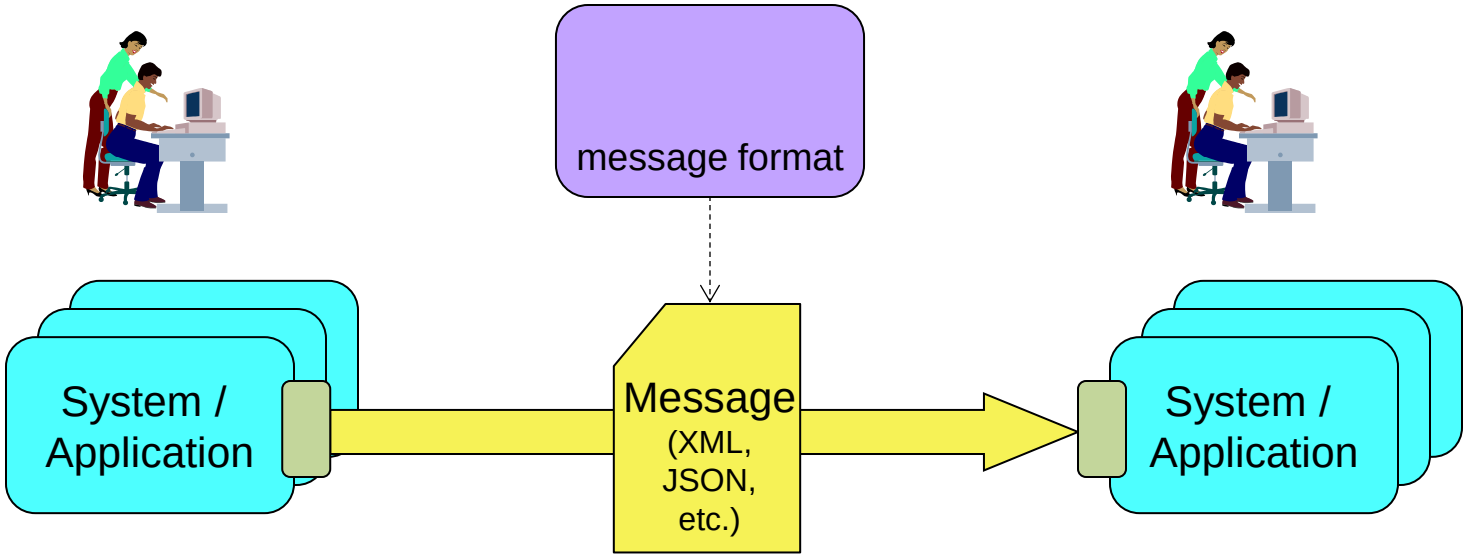


NIEMOPEN

IMPLEMENTING THE EXCHANGE SPECIFICATION

Other producers and consumers may join at any time, in any order, by implementing the message format.

By following the NIEM standards-based approach, machine-to-machine data exchanges can be implemented in less time and at lower cost for the enterprise as a whole



NIEMOPEN

NIEM MESSAGE DATA IN RDF

- **NIEM messages can be converted into RDF**
  - The NTAC is providing free and open-source tools for these conversions

```
<ex:CrashDriverInfo>
 <j:Crash>
  <j:CrashVehicle>
   <j:CrashDriver>
    <nc:Person s:id="P1">
     <nc:PersonBirthDate>
      <nc:Date>1890-05-04</nc:Date>
     </nc:PersonBirthDate>
     <nc:PersonName>
      <nc:PersonGivenName>Peter</nc:PersonGiv
      <nc:PersonMiddleName>Death</nc:PersonMi
```

```
_:n0 a j:CrashType ;
     j:CrashVehicle _:n1 .
_:n1 a j:CrashVehicleType ;
     j:CrashDriver _:n2 .
_:n2 a j:CrashDriverType ;
     nc:Person _:P1 ;
_:P1 a nc:PersonType ;
     nc:PersonBirthDate _:n3 ;
     nc:PersonName _:n4 .
_:n3 a nc:DateType ;
     nc:Date "1890-05-04" .
```

NIEM-based message in XML ⟵ *equivalent to* ⟶ NIEM-based message in RDF

NIEMOPEN

NIEM MODELS IN RDF

- **NIEM models can be converted into RDF**
  - The NTAC is providing free and open-source tools for these conversions

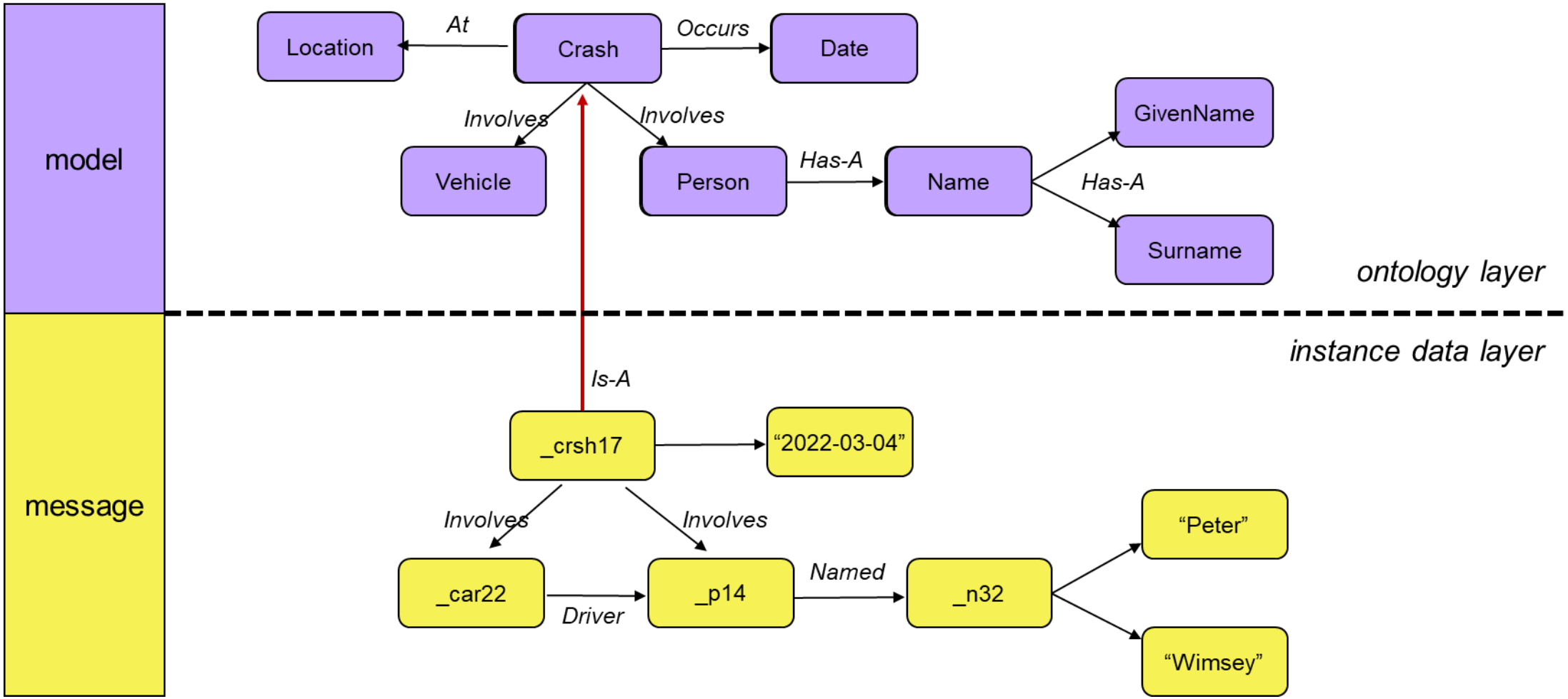
```
<xs:complexType name="PersonType">
 <xs:annotation>
  <xs:documentation>A data type for a human being.<
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="structures:ObjectType">
   <xs:sequence>
    <xs:element ref="nc:PersonBirthDate"/>
    <xs:element ref="nc:PersonName"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

```
nc:PersonType
 a owl:Class ;
 rdfs:comment "A data type for a human being." .

nc:PersonName
 a owl:ObjectProperty ;
 rdfs:domain nc:PersonType ;
 rdfs:range nc:PersonNameType ;
 rdfs:comment "A combination of names and/or
titles by which a person is known." .

nc:PersonBirthDate
    a owl:ObjectProperty ;
```

NIEM model in XSD ⟵ *equivalent to* ⟶ NIEM model in RDF

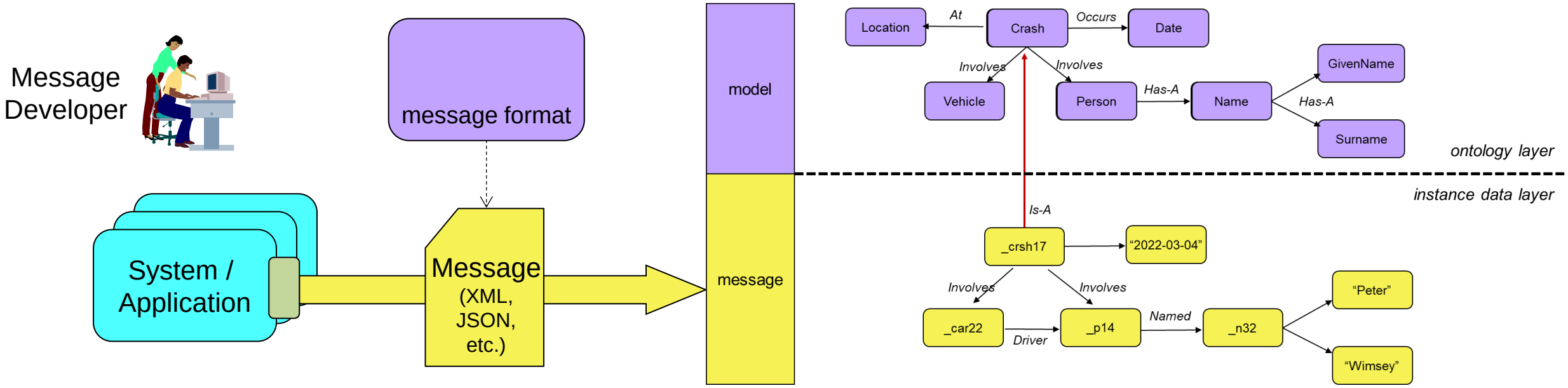
NIEMOPEN

MODEL + MESSAGE = KNOWLEDGE GRAPH



NIEMOPEN

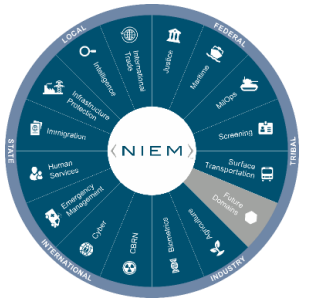
NIEM DATA AS A KNOWLEDGE GRAPH



NIEMOPEN

MODEL + MESSAGE = KNOWLEDGE GRAPH

- **Data from systems that generate NIEM-based messages is available as a knowledge graph**
  - With no extra effort from the developers of those systems
- **If the NIEM model becomes aligned with useful upper ontologies, then NIEM message models and instance data will also be aligned**
- **This could reduce your data wrangling workload even more**



NIEMOPEN

RELATING A NIEM MODEL TO CCO

- Relationships to upper / middle ontologies can be added now
- Over time, the NIEM model could change to align with ontologies

```
<xs:complexType name="PersonType">
 <xs:annotation>
  <xs:documentation>A data type for a human being.<
 </xs:annotation>
 <xs:complexContent>
  <xs:extension base="structures:ObjectType">
   <xs:sequence>
    <xs:element ref="nc:PersonBirthDate"/>
    <xs:element ref="nc:PersonName"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

```
nc:PersonType
 a owl:Class ;
 rdfs:comment "A data type for a human being." .

nc:PersonName
 a owl:ObjectProperty ;
 rdfs:domain nc:PersonType ;
 rdfs:range nc:PersonNameType ;
 rdfs:comment "A combination of names and/or
titles by which a person is known." .

nc:PersonType rdfs:subclassOf cco:Person .
```

NIEM model in XSD ⟵ *equivalent to* ⟶ NIEM model in RDF

NIEMOPEN

A CO-OPERATOR, NOT A COMPETITOR

A new slide, based on questions at the summit

- **Are you an ontologist?**
  - We aren't trying to get *you* to use NIEM
  - We aren't trying to replace what you are doing
- **There are many source systems exposing data through an API, today and tomorrow**
  - We want them to use NIEM
  - Then they will reuse community definitions, instead of inventing everything from scratch
  - And their data will be available as a knowledge graph
- **We want you to help us do that better**
  - So that their data will be more useful to you
  - So that we can help them reuse the knowledge in your ontologies

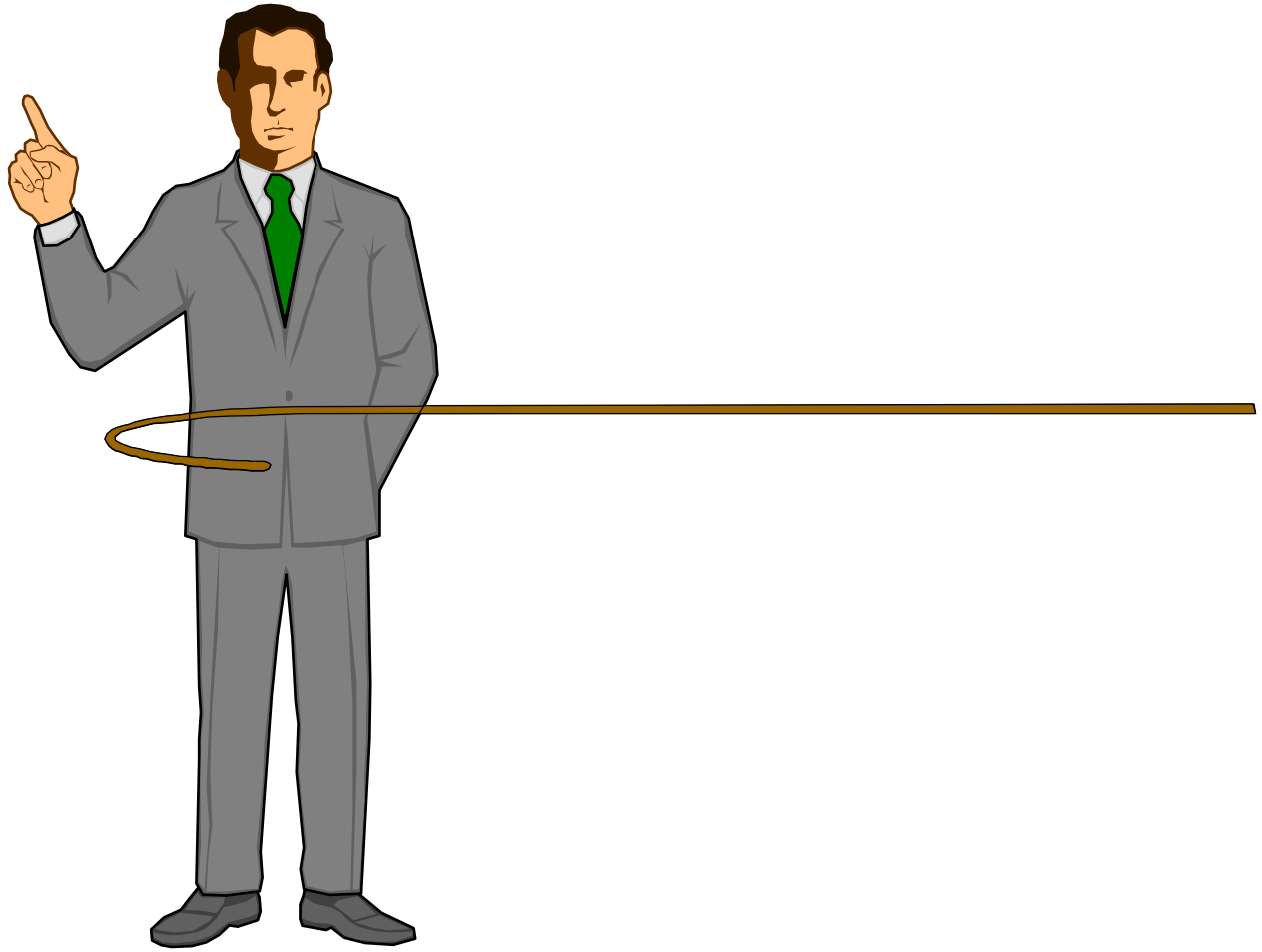
NIEM AND ONTOLOGIES: THE FUTURE

- **NIEM is an ontology player, not a competitor**
  - Designing data contracts means NIEM will always be a little different, but...
  - Will gradually converge, as NIEM shifts from its XSD beginning to its RDF foundations
- **Already seeing some exciting applications:**
  Sonoma County CA criminal justice
  - Multiple sources reconciled through a NIEM model
  - Data from sources converted to RDF in a COTS triple store
  - LLM uses NIEM model to generate SPARQL query explanations
- **Many sources will be available as knowledge graphs**
  - NIEM as an "ontology on-ramp" – will reduce data wrangling effort
- **Next step: Explore alignment with BFO/CCO**
  - Small-scale experiment to demonstrate value

Dr. Scott's Crystal Ball

NIEMOPEN

SUMMARY



SUMMARY

- **NIEM is a framework for designing machine-to-machine data exchange specifications**
  - A technical architecture for message models and runtime messages
  - A library of community-agreed data definitions: the NIEM Model
- **Many NIEM-based data exchanges in operation today**
- **In NIEM 6.0, models and messages can be converted to RDF**
  - With no extra effort from message designers and developers
  - The result is a knowledge graph (at least in form)
  - Could provide data for inferencning and SPARQL queries w/o much data wrangling; however...
- **The NIEM model today is a sparse ontology**
  - No relations to any upper ontology at present
  - Adding those relations will take some work
  - Need a demonstration project to show the value

RESOURCES

- **NIEM Technical Architecture Committee**
  - Co-chair: Scott Renner, sar@mitre.org
  - Github: github.com/niemopen/ntac-admin
- **NIEMOpen Managing Director:** Katherine Escobar, katherine.b.escobar.civ@mail.mil
- **Useful documents, specifications, tools**
  - *Understanding the NIEM Technical Architecture*
    https://github.com/niemopen/ntac-admin/tree/main/project-notes
  - *NIEM 6.0 Naming and Design Rules*
    https://github.com/niemopen/niem-naming-design-rules/tree/dev
  - *NIEM Common Model Format (CMF) specification*
    https://github.com/niemopen/common-model-format
  - *Common Model Format Tool (CMFTool)*
    https://github.com/niemopen/cmftool