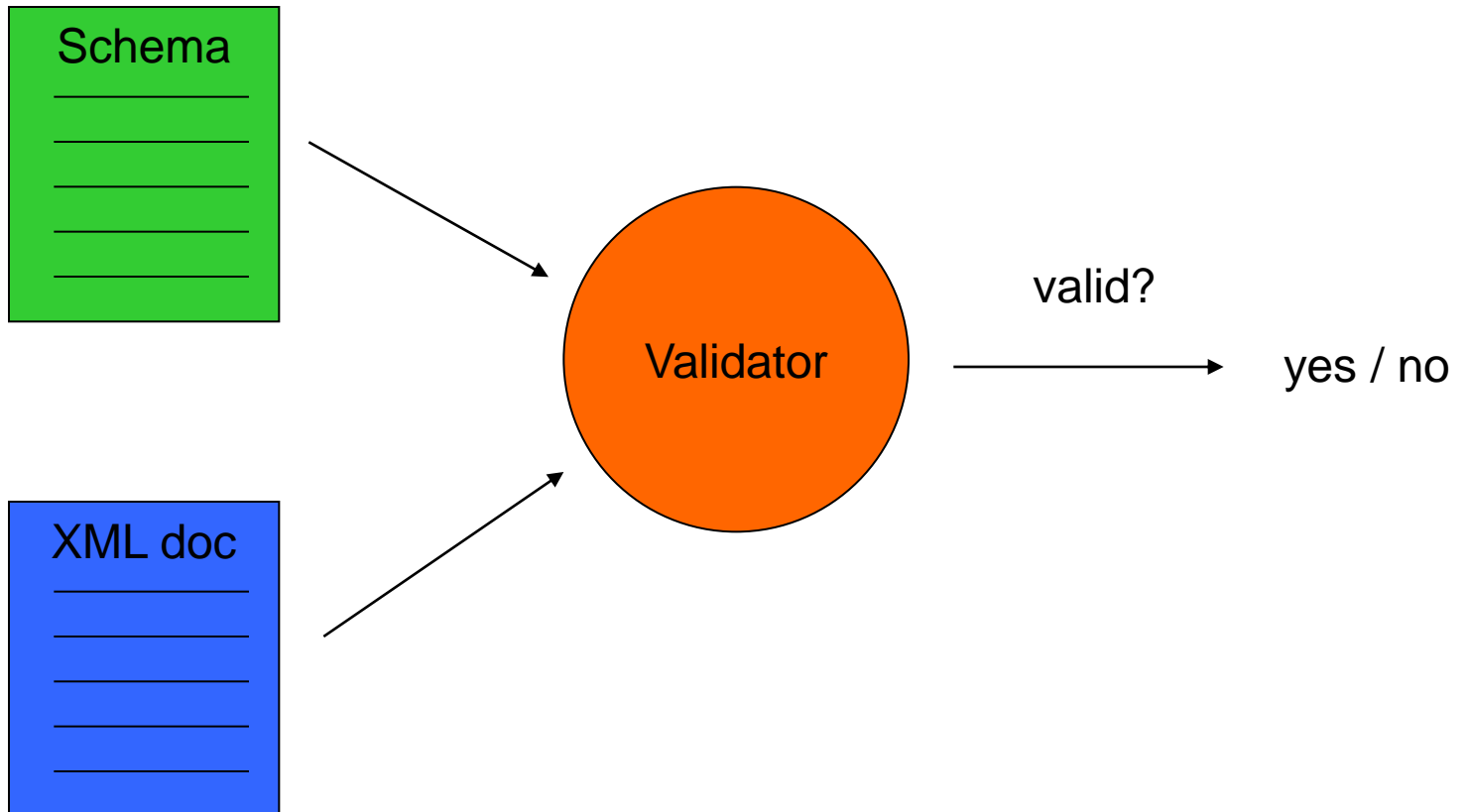


# Schema Issues

# XML Schema



# Schema's serve multiple masters

- Schema's serve more than one purpose
  - design
  - validation
  - contract
  - code generation
- those purposes often need different Schema's

# Schema's serve multiple masters

- design
  - + reuseability, composability, simplicity
  - performance
- validation
  - + performance, strictness, error messages, completeness
  - reuseability, composability, simplicity, readability
- contract
  - + readability, strictness, completeness
  - performance
- code generation
  - + simplicity, readability
  - reuseability, composability

# The HL7v3 Schema's

- Let's look at an example
- Get Person Demographics Query
- Send in person id
- Get name, address, birthdate et cetera

# The HL7v3 Schema's

```
<?xml version="1.0" encoding="UTF-8"?>
<QUPA_IN101101 xmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-in:
  <id extension="700804" root="2.16.528.1.1007.3.3.304845.2"/>
  <creationTime value="20061127140000"/>
  <versionCode code="NICTIZED2005-Okt"/>
  ...
  <ControlActProcess moodCode="EVN">
    <effectiveTime value="20061127140000"/>
    <authorOrPerformer typeCode="AUT">
      <participant>
        <AssignedPerson>
          <id extension="000120450" root="2.16.528.1.1007.3.1"/>
          ...
        </AssignedPerson>
      </participant>
    </authorOrPerformer>
    <queryByParameter>
      <queryId extension="6604" root="2.16.528.1.1007.3.3.304845.5"/>
      <responseModalityCode code="R"/>
      <statusCode code="executing"/>
      <person.id>
        <value extension="100197231" root="2.16.840.1.113883.2.4.6.3"/>
      </person.id>
    </queryByParameter>
  </ControlActProcess>
</QUPA_IN101101>
```

# The HL7v3 Schema's

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- BSN (SBV-2) specifieke payload -->
<xs:schema targetNamespace="urn:hl7-org:v3" xmlns="urn:hl7-org:v3" xmlns:hl7="urn:hl7-
  <xs:include schemaLocation="../coreschemas/datatypes.xsd"/>
  <xs:include schemaLocation="../coreschemas/voc.xsd"/>
  <xs:include schemaLocation="../schemas/MCCI_MT000100.xsd"/>
  <xs:include schemaLocation="../schemas/QUQI_MT021001.xsd"/>
  <xs:include schemaLocation="../schemas/QUPA_MT101101_BSN.xsd"/>
  <xs:element name="QUPA_IN101101" type="MCCI_MT000100.Message"/>
</xs:schema>
```

# The HL7v3 Schema's

QUPA\_101102\_V01

- MCCI\_MT000300UV01
  - COCT\_MT040203UV01
    - COCT\_MT150003UV03
    - COCT\_MT030203UV02
- MFMI\_MT700711
  - COCT\_MT090300UV01
    - COCT\_MT150000UV02
      - COCT\_MT070000UV01
        - » COCT\_MT710000UV01
    - COCT\_MT150003UV03
    - COCT\_MT070000UV01
      - COCT\_MT710000UV01
  - COCT\_MT090100
    - COCT\_MT150000UV02
      - COCT\_MT070000UV01
        - » COCT\_MT710000UV01
    - COCT\_MT150003UV03
    - COCT\_MT070000UV01
      - COCT\_MT710000UV01
  - COCT\_MT090003
    - COCT\_MT150003UV03
  - MCAI\_MT900001
- QUPA\_MT101102\_V01
- QUPA\_MT101101\_V01



# The HL7v3 Schema's

- The XML document, though abbreviated, isn't difficult
  - (SOAP omitted here...)
  - Transmission Wrapper: message-id, creation date
  - Act Wrapper: query issuer etc.
  - Payload: person-id
- The Schema is very simple
  - 5 includes and 1 element
  - but not very readable!
  - the schema inclusion tree is very complex

# The HL7v3 Schema's

- Schema's should be readable
  - tools can solve this
  - but they make you dependent on the tool
- Therefore: flatten the Schema's
  - remove all includes
  - put included schema's where they belong
- For readability: make the Schema resemble the instance
- Readable Schema's generate readable code!

# Flatten the Schema's

# The HL7v3 Schema's

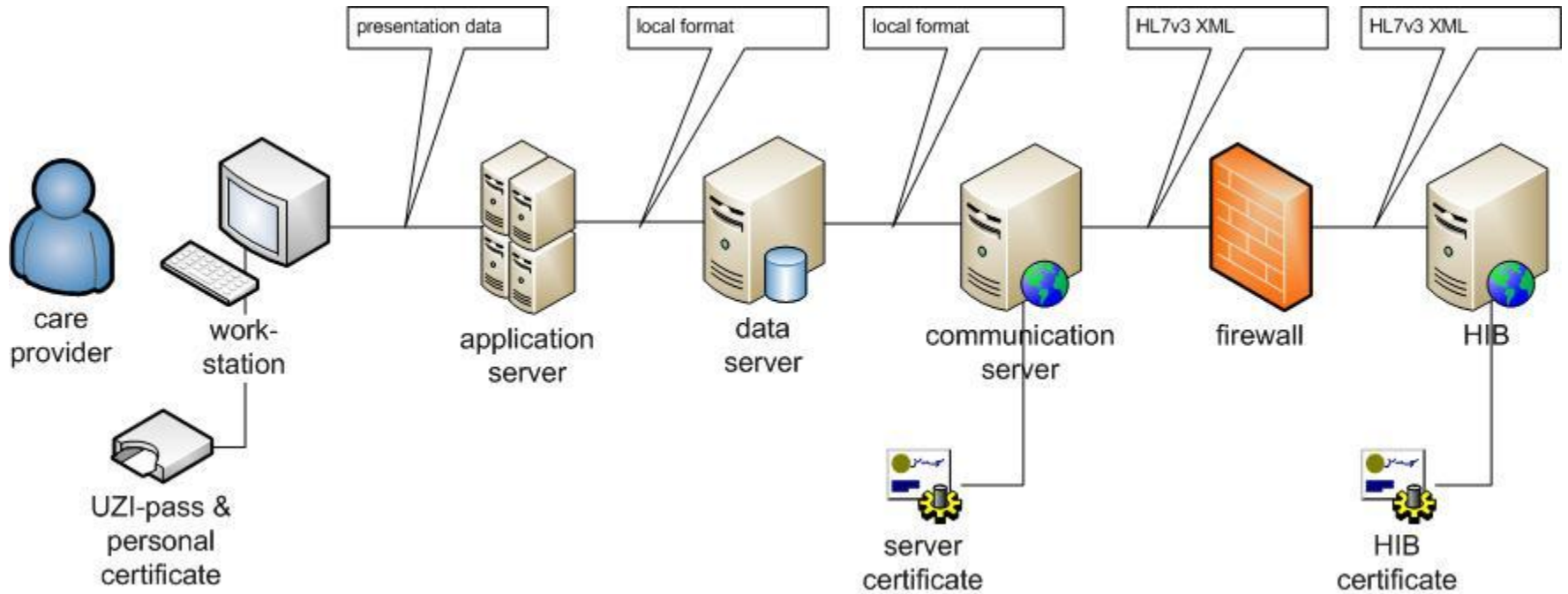
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema targetNamespace="urn:hl7-org:v3" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:hl7="urn:hl7-org:v3" >
  <xs:include schemaLocation="../coreschemas/datatypes.xsd"/>
  <xs:include schemaLocation="../coreschemas/voc.xsd"/>
  <xs:include schemaLocation="COCT_MT040203.xsd"/>
  <xs:element name="Message" type="MCCI_MT000100.Message"/>
  <xs:group name="MCCI_MT000100">
    <xs:sequence>
      <xs:element name="Message" type="MCCI_MT000100.Message"/>
    </xs:sequence>
  </xs:group>
  <xs:complexType name="MCCI_MT000100.Message">
    <xs:sequence>
      <xs:element name="id" type="II" maxOccurs="unbounded"/>
      <xs:element name="creationTime" type="TS" maxOccurs="unbounded"/>
      <xs:element name="versionCode" type="CS" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="interactionId" type="II" maxOccurs="unbounded"/>
      <xs:element name="profileId" type="II" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="processingCode" type="CS" maxOccurs="unbounded"/>
      <xs:element name="processingModeCode" type="CS" maxOccurs="unbounded"/>
      <xs:element name="acceptAckCode" type="CS" maxOccurs="unbounded"/>
      <xs:element name="attentionLine" type="MCCI_MT000100.AttentionLine" minOccurs="1" maxOccurs="unbounded"/>
      <xs:element name="receiver" type="MCCI_MT000100.Receiver"/>
      <xs:element name="respondTo" type="MCCI_MT000100.RespondTo" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="sender" type="MCCI_MT000100.Sender"/>
      <xs:group ref="ControlAct"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

# The HL7v3 Schema's

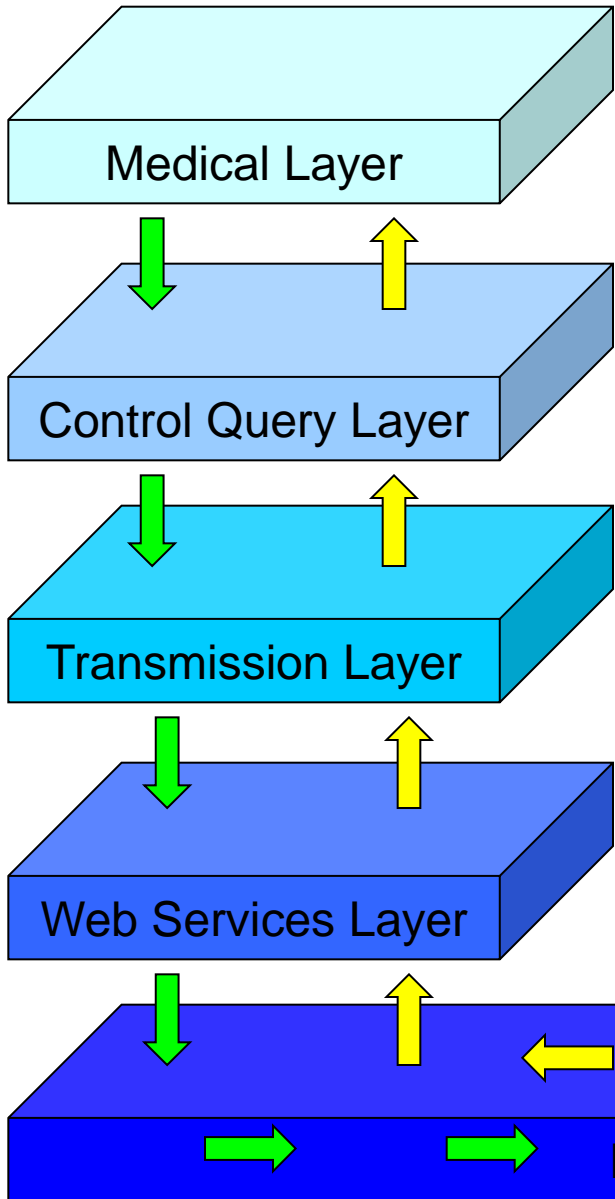
- HL7 datatypes
  - TS: Point in Time
  - CS: Simple Coded Value
  - ST: Character String
- Translate to XSD
  - datetime, string
- HL7 datatypes predate XSD datatypes
- With a lot of HL7 datatypes, nothing happens except translation to XSD datatypes
- Do this in the source, generates much more readable code

# Simplify the Schema's

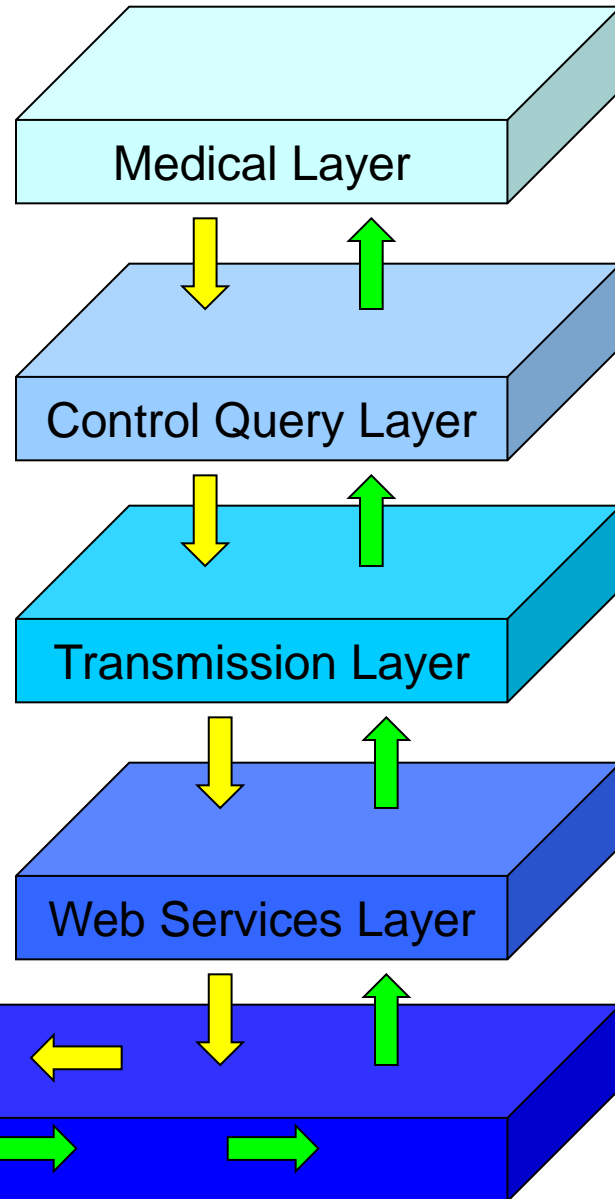
# Layering



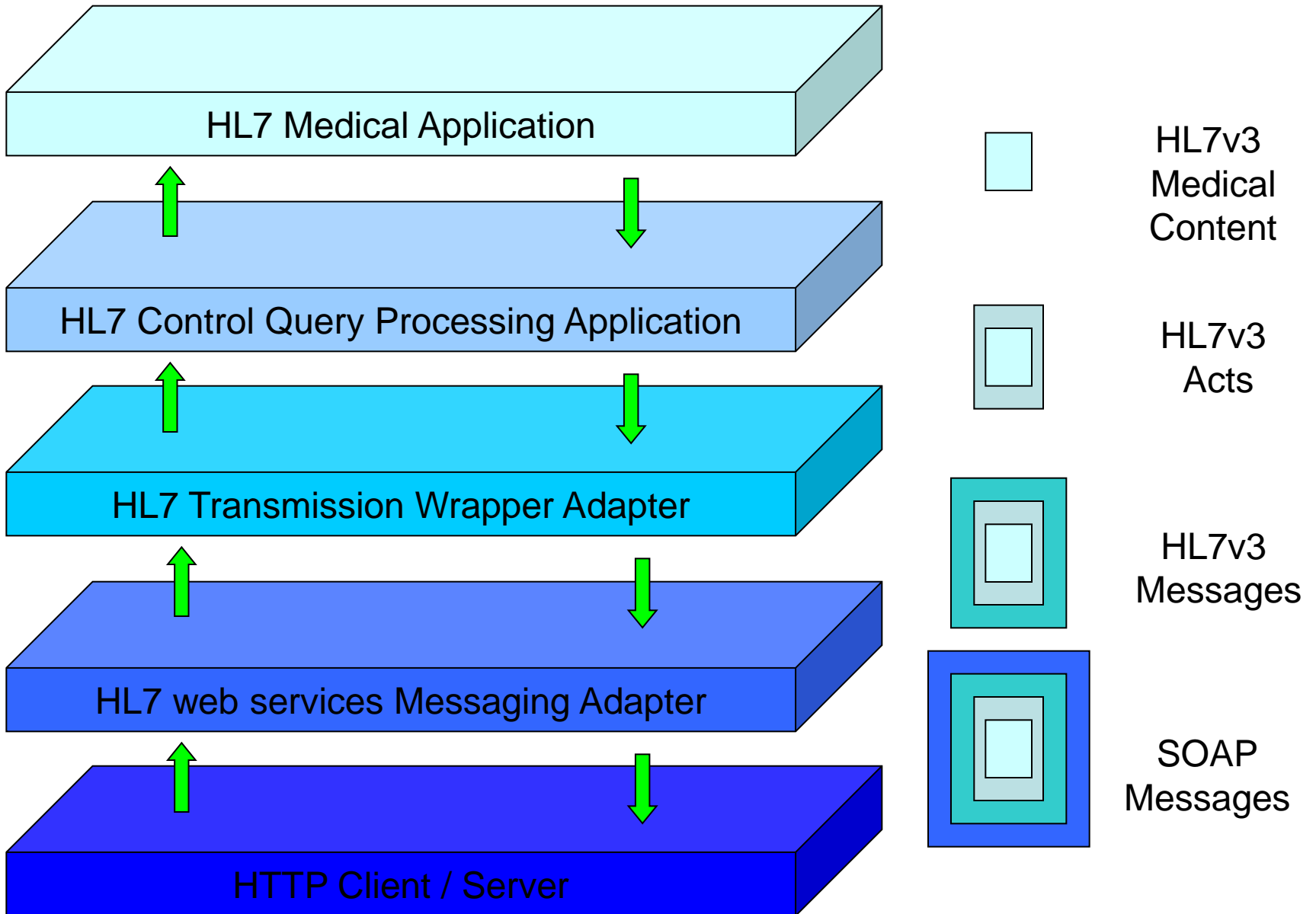
# Initiating Application



# Responding Application

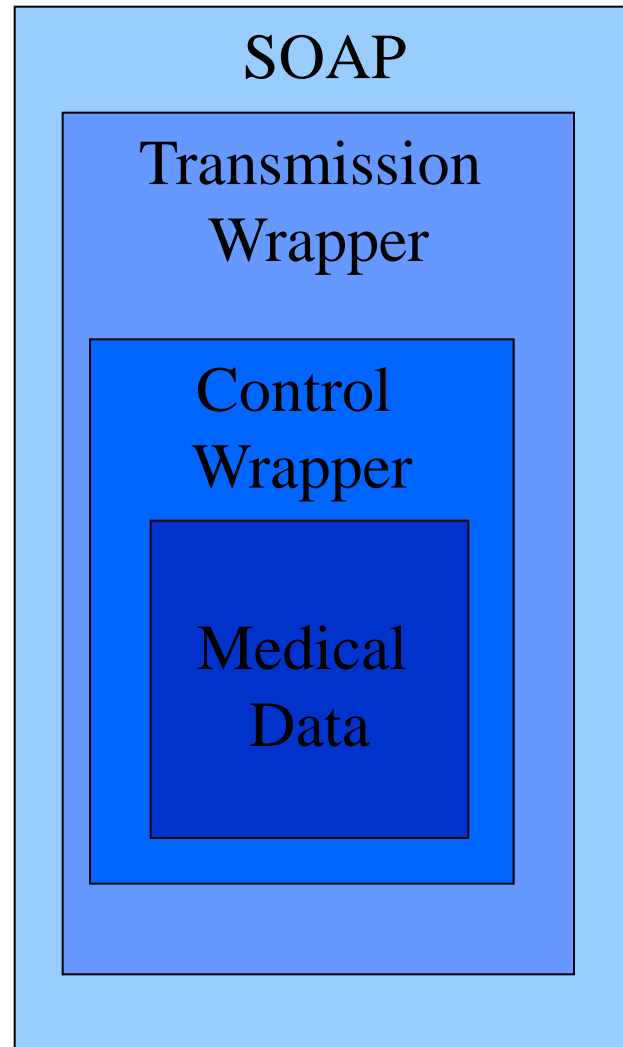


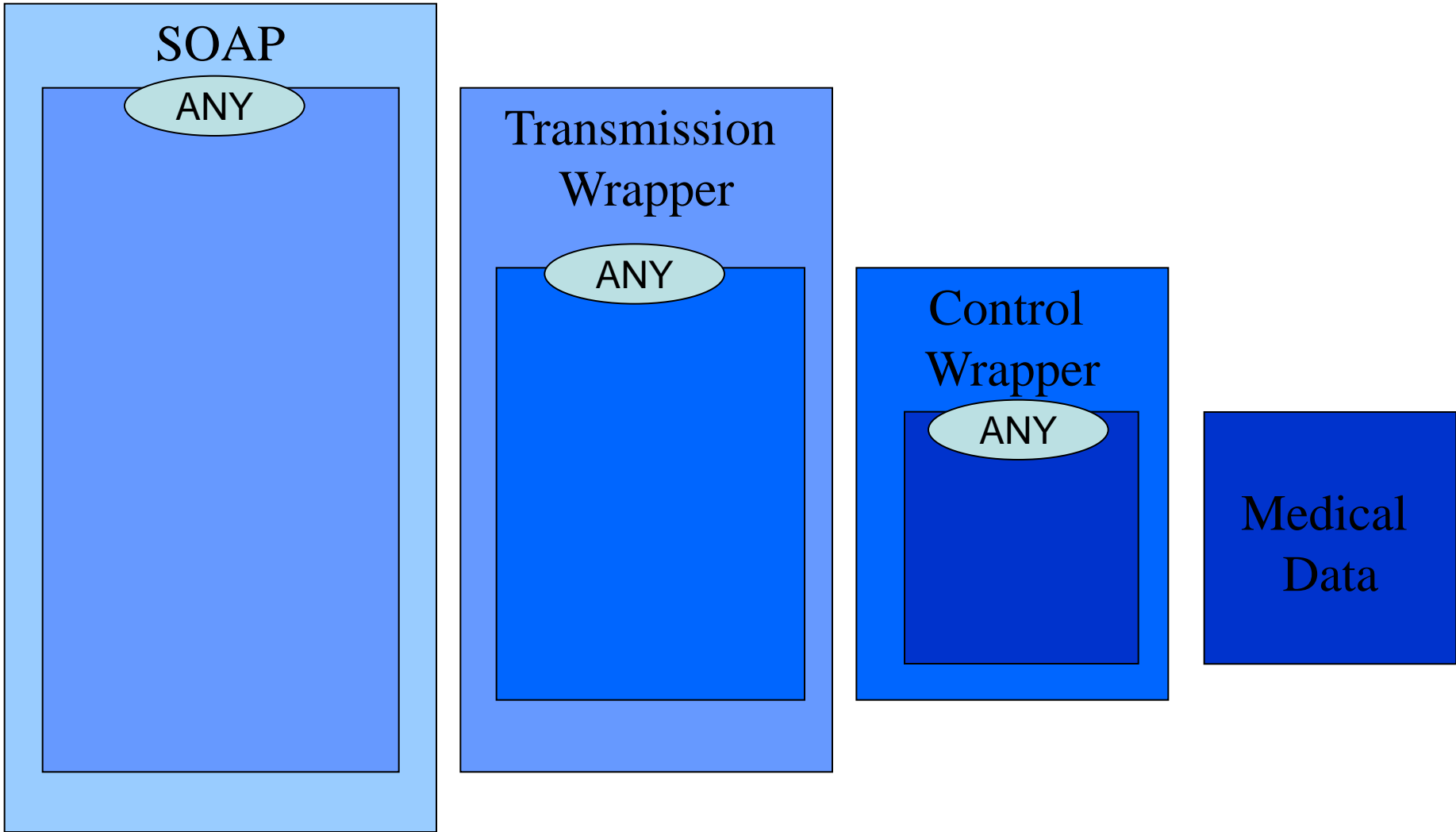




# The HL7v3 Schema's

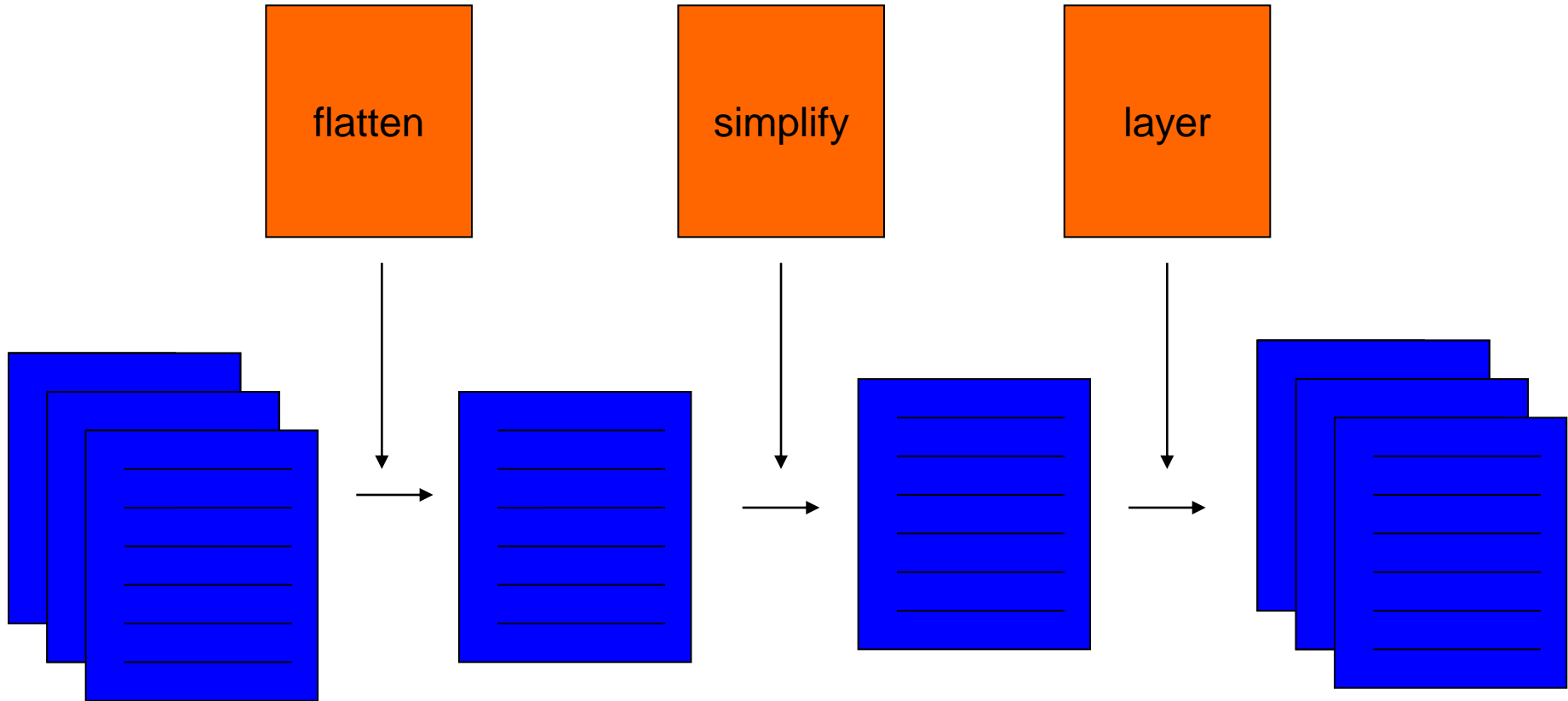
- layer the Schema's
- anonymize with xs:any
  - `<xs:any namespace="##any" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>`
- SOAP Headers, soap:Body anon
- HL7 Transmission Wrapper, Act anon
- HL7 Act Wrapper, medical payload anon





# Layer the Schema's

# The HL7v3 Schema's



# The HL7v3 Schema's

- Not very readable
  - without Schema editor not practically feasible
- Generated from database
- Fix them with XSLT or other:
  - Flatten the Schema's
  - Remove unneeded datatype hierarchies
  - Layer the Schema's
- Makes the Schema's much more readable
- Generates simpler code
- New ITS (HL7 Schemaset) is coming, but the approach sketched here will probably remain valid

# The HL7v3 Schema's

James Clark:

“validity is a relationship between a document and a schema, not a property of a document”



# The HL7v3 Schema's

schema's can be equivalent:  
when two schema's consider  
the same set of documents valid  
the schema's are equivalent

# The HL7v3 Schema's

don't think of THE schema,  
but the SCHEMAS

VARIANTS

# The HL7v3 Schema's

