The Integrated Data Environment

HL7 And Beyond

A. Stewart Ferguson, Ph.D.
Director, AFHCAN

Oran Dennison
Sr. Software Architect

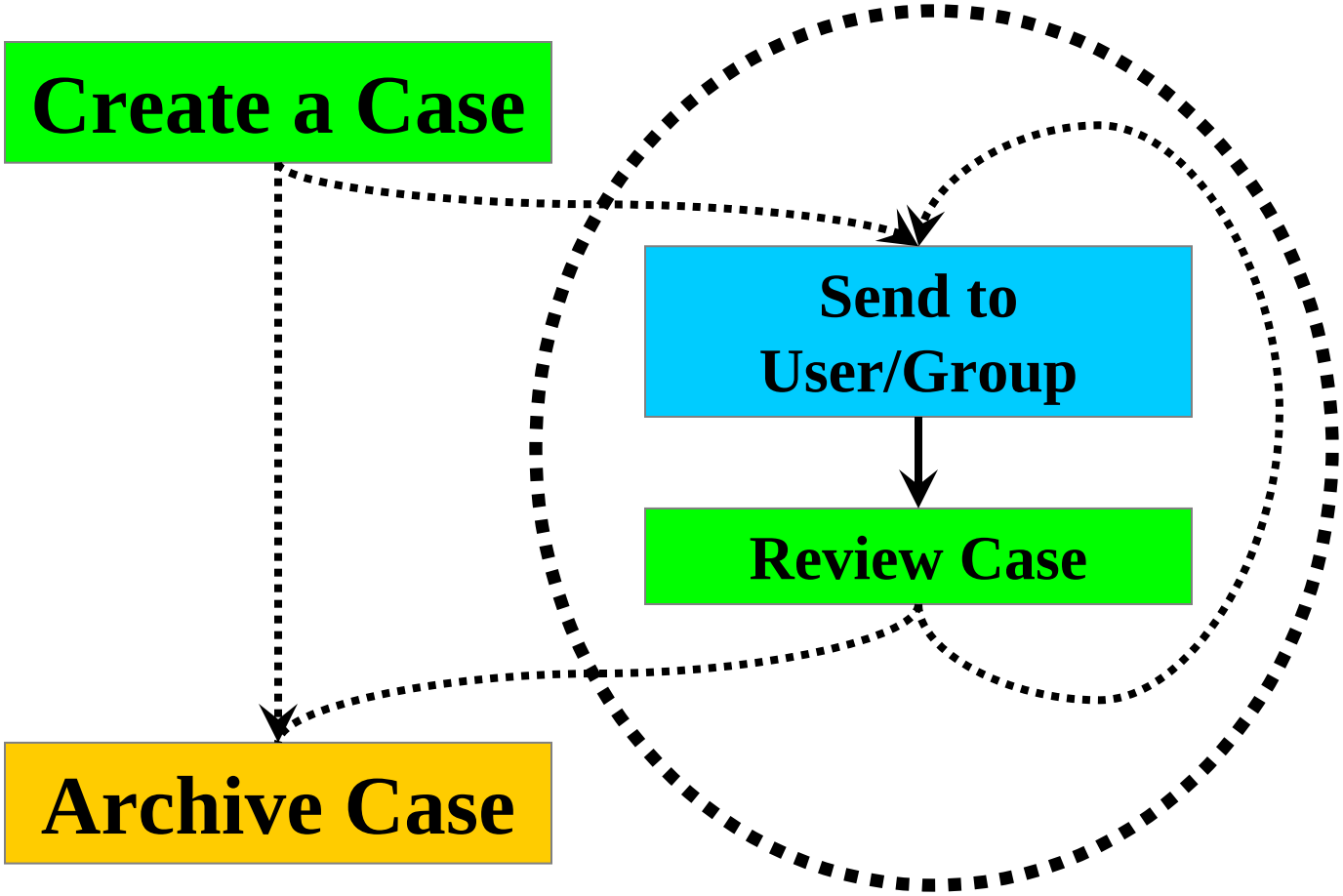
Alaska Native Tribal Health Consortium
Anchorage, AK, USA

A Telehealth Case
"Encounter centric"

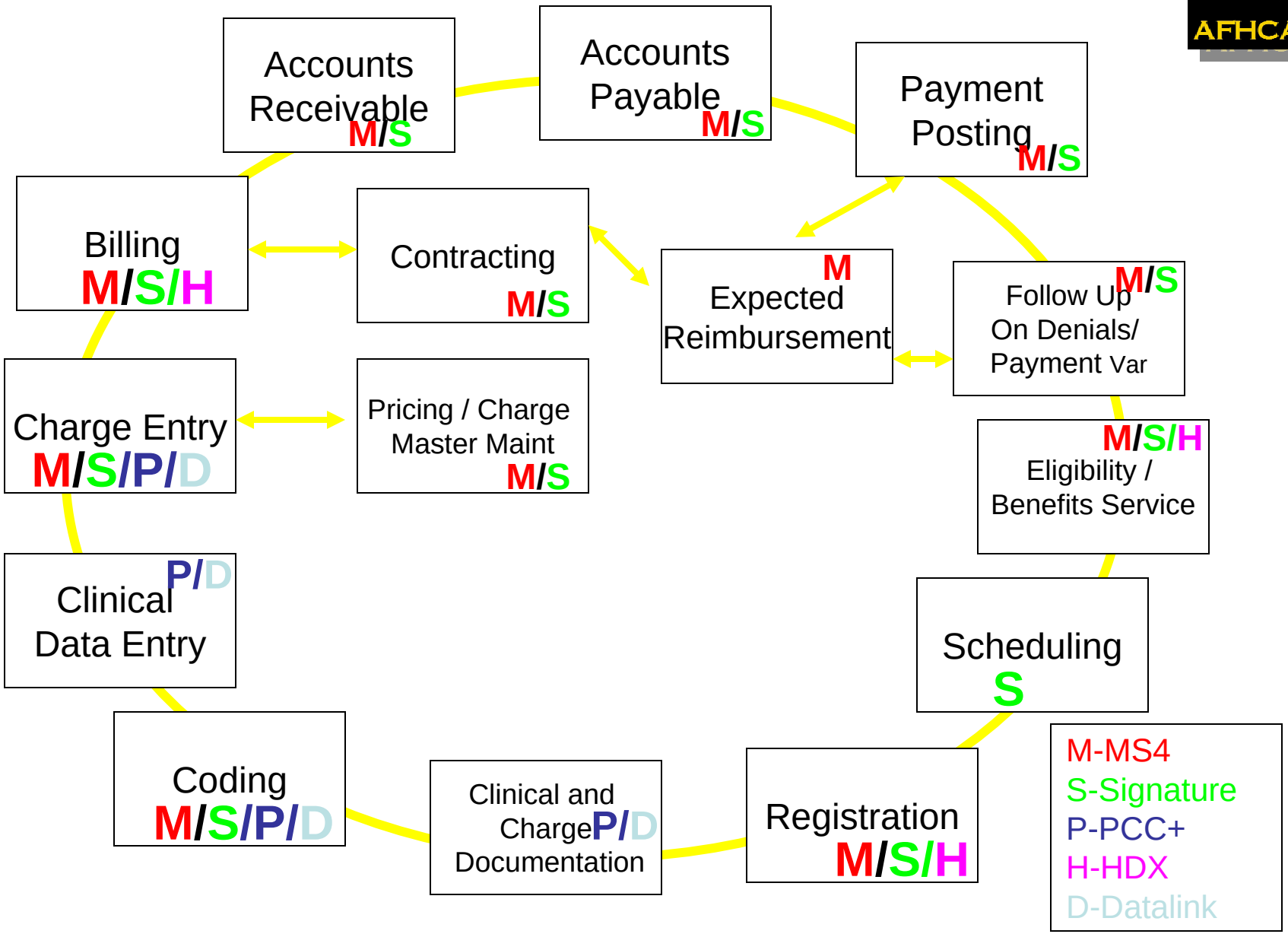
- A "Case" is created for a single patient – typically for a single visit.

- *Test* versus *Real* cases
  - Different patient databases
  - Eval questions only asked on real cases
  - Clearly identified on screen with colorful header

The LIFE CYCLE of an AFHCAN Case.



ANMC Revenue Cycle 2000-2005



RPMS Integration Use cases

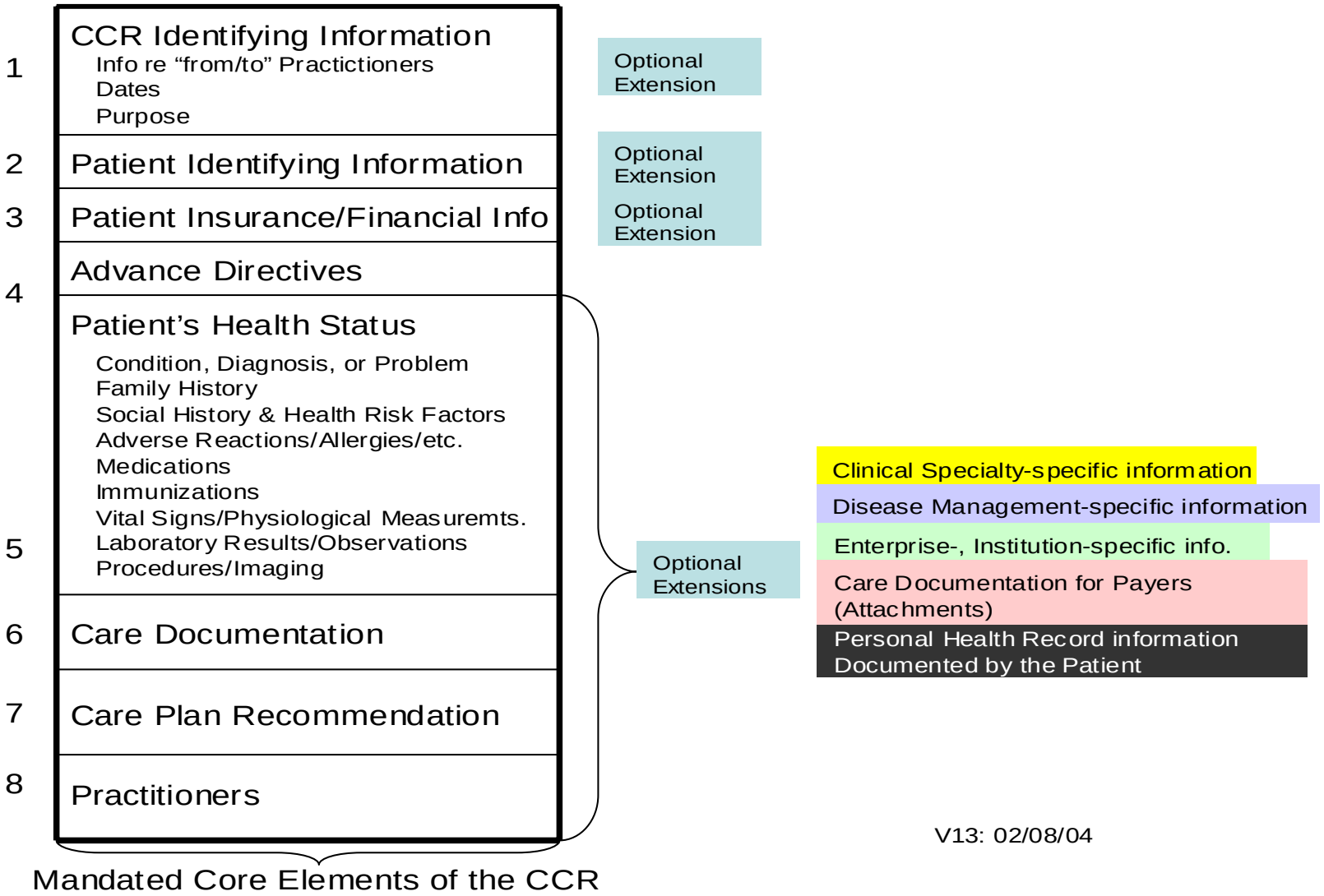
- Get patient demographics from RPMS
- Tell RPMS about AFHCAN case archives
- Retrieve health summaries from RPMS
- Patient query solutions
- Send cases from RPMS to AFHCAN
- Send cases from AFHCAN to RPMS
- Interface to binary data repositories

What “additional” information does telehealth need?

- Multiple candidates
  - Related clinical events
  - Patient history
  - Labs
  - Immunizations
  - Allergies

- How is this obtained? CCR, CDA, …

What's in the CCR Data Set?



V13: 02/08/04

Implementation

- Issues creating a spec
- Implementation technologies
- Issues implementing a spec
- How AFHCAN integrates with AFHCAN
- Current integration trends applied to HL7

Issues creating a spec

- Coming to a common understanding
- Defining deviations from “standard” HL7
- Reference data (coding systems, etc.)
- “Must understand”, “must echo”, hard-code
- Semantic differences
- Generic vs. site-specific interfaces

It takes two (or more) to tango

- Agree on use cases, message exchanges, message structure, and more (conformance profile ++)
- There is no substitute for getting the geeks talking to each other

“Non-standard” HL7 segments

- Missing MSH fields
- Patient information in the wrong PID fields
- Z-segments
- What else is lurking in the shadows?

The problem of “reference data”

- Immutable data is fine
- Versioned data is fine
- Only having a version-independent identifier has unpredictable results
- Master file synchronization

How much must be "understood"?

- What does RPMS need echoed to maintain internal correlation?
- How much of the various coding systems do we need to understand or hard-code?

Semantic differences

- Patient-centric vs. encounter-centric
- What is a visit? A consult? A diagnosis? A billable event?
- One-to-many differences
  - A single telehealth case may be composed of multiple sessions with input from multiple providers
- Can a case be re-opened?

Generic solution vs. a site-specific solution

- Highly unlikely that a site-specific solution will work for all RPMS installations
- IHS-wide solution has more to consider
  - Viewing telehealth data in EHR
  - Compatibility with multiple RPMS versions
  - Independent patch schedules

Issues implementing a spec

- Technology choices
- Data storage and mapping
- Communication
- Maintenance

Implementation technologies

- On the AFHCAN side
  - Chameleon
- On the RPMS side
  - GIS
  - BMX + HL7 for query?

Chameleon

- Cross-platform HL7 messaging toolkit
- Multi-language
- Customizable to any bizarre HL7 format
- Drag-and-drop mapping
- XML conversion
- Shields you from site-specific changes

iNTERFACEWARE Chameleon - Untitled.vmd - [Message Browser]

File Edit View Insert Chameleon Window Help

New... Open... Save | default (Complete) | Message Table Segment Composite DateTime | Options... Stubcode Browse

- Batch Header and Footer Segments
- Messages
  - Message1 ()
    - Segment Grammar
    - Table Grammar
    - Identity()
  - Message2 ()
    - Segment Grammar
    - Table Grammar
    - Identity()
- Tables
- Segments
  - MSH (Message Header)
  - PID (Patient Identification)
- Composites
  - CE (Coded Element)
  - CWE (Coded With Exceptions)
  - CX (Extended Composite ID With C
  - DLN (Driver_s License Number)
  - DR (Date/time Range)
  - DT (Date)
  - DTM (Date/time)
  - EI (Entity Identifier)
  - FN (Family Name)
  - HD (Hierarchic Designator)
  - ID (String Data)
  - IS (String Data)
  - MSG (Message Type)
  - NM (Numeric)
  - PT (Processing Type)
  - SAD (Street Address)

```
MSH|^~\&|MESA_ADT|XYZ_ADMITTING|MESA_IS|XYZ_HOSPITAL
EVN||200004211000||||200004210950
PID|||583020^^^ADT1||WHITE^CHARLES||19980704|M||AI|7
PV1||E||||||5101^NELL^FREDERICK^P^^DR||||||||||V10C
```

Parse

Complete Parse

☐ Show Field Number

- Message - Unrecognized
  - MSH - Message Header
    - Field Separator("|")
    - Encoding Characters("^~\")
    - Sending Application("MESA_ADT")
    - Sending Facility("XYZ_ADMITTING")
    - Receiving Application("MESA_IS")
    - Receiving Facility("XYZ_HOSPITAL")
    - Date/time Of Message("")
    - Security("")
    - Message Type
      - Message Code("ADT")
      - Trigger Event("A04")
    - Message Control ID("101102")
    - Processing ID("P")
    - Version ID("2.3.1")
  - EVN||200004211000||||200004210950
  - PID|||583020^^^ADT1||WHITE^CHARLES||19980704|M||AI|7616 STANFORD AVE^^ST. LOUIS^MO^6313
  - PV1 - Undefined Segment
    - Field 0("")
    - Field 1("E")
    - Field 2("")
    - Field 3("")
    - Field 4("")
    - Field 5("")

BMX possibilities

```
// Code for mapping Chameleon HL7 patient data
// to AFHCAN Patient object
Patient p = new Patient();
p.RegionalHR = msg.Patient(0).ID;
p.FirstName = msg.Patient(0).FirstName;
p.LastName = msg.Patient(0).LastName;
p.DOB = msg.Patient(0).DOB.DateTimeValue;
p.Sex = msg.Patient(0).Sex;
p.SSN = msg.Patient(0).SSN;
p.City = msg.Patient(0).City;
p.State = msg.Patient(0).State;
p.HomePhone = msg.Patient(0).HomePhone;
p.WorkPhone = msg.Patient(0).WorkPhone;
// . . . fill in additional values
databaseManager.StorePatient(p);
```

Swap sides and replace Patient with ADO.NET DataTable and databaseManager.StorePatient with llpServer.SendMessage

New data needs a home

- RPMS development to support telehealth data
  - VConsult file?
  - TIU notes?
- Define mappings to HL7

Maintenance duties

- People to work the error queues on a daily basis
- Patient merging/linking
- Coordinating master file updates and versioning
- Responding to RPMS patches
- AFHCAN adds new types of data

How AFHCAN interfaces with AFHCAN

- “Self-integration” issues
  - Distributed across many autonomous organizations
  - Independent upgrade schedules
  - Control over who shares what with whom
  - Cold Fusion to .NET
  - Almost 50 servers talking to each other

“Self-integration” solutions

- XML messaging
- Standards-based security
  - 3DES encryption and digital signatures
  - Message-level security, not just transport-level
- Transport-neutral messaging
  - Firewall-friendly intermediary message routers
    - Hub-and-spoke model scales better than point-to-point
  - Transport over HTTP, HTTPS, TCP, and more
- Map or block breaking changes
- Reliable store-and-forward queuing protocol

AFHCAN integration with other systems

- HL7
  - v2.x
  - v3?
- DICOM
- XML web services
- Other?

How other systems integrate

- XML web services
  - Basic SOAP messaging
  - WS-* (used by HL7 v3 Web Services Profile)
    - WS-Security
    - WS-Addressing
    - WS-ReliableMessaging
    - WS-Policy
    - WS-SecurityPolicy
    - WS-Trust
    - WS-SecureConversation

HL7 and SOA

- Service Oriented Architecture
  - Service-Orientation
  - Architecture
- Service-orientation can be applied to v2, but it’s hard
- Ongoing HL7 activity around SOA for v3

4 Tenets of SOA

- Boundaries are explicit
- Services are autonomous
- Services share schema and contract, not implementation
- Service compatibility is based on policy

1: Boundaries are explicit

- Internal implementation details shouldn't leak outside the service boundary
  - Say no to RPC-style interfaces
  - Say no to shared-database integration
- Services expose coarse-grained operations at the logical business layer
  - Self-contained, meaningful message context
  - Chunkier, less chatty messaging
    - Unsolicited Observation messages - patient, visit, order, observation
    - HL7 support for batch message transfer

2: Services are autonomous

- Services are deployed and versioned independently of each other
  - Can be rewritten without negatively impacting partners
  - “Conform to the HL7 message profile, and everything will work out fine”
- Services are built to last, service configurations are built to change
  - New clients, transports, security
  - Backward-compatible mappings

2: autonomy cont'd.

- Systems may not always be reliably connected
  - Queueing, reliable delivery mechanisms
  - Master files and caching as fallback for query
- No CRUD-y interfaces!
  - CRUD doesn't reflect the system's business rules
  - Doesn't enable independent evolution
  - HL7's use-case-driven message exchanges
    - ADT feeds are CRUD with business semantics
    - Master files are CRUD-oriented reference data
    - V3's Message Development Framework

3: Services share schema and contract, not implementation

- HL7 nails this!
- Shared schema = HL7 static definition
- Shared contract = HL7 dynamic definition
- If a schema or contract requires a breaking change, version the service.
  - Incrementing HL7 version numbers
  - Implement multiple versions of HL7
    - Mapping tools help shield you from differences

4: Service compatibility is based on policy

- Domain-specific policy assertions ensure semantic compatibility
- Security policies
  - “I decline to talk to you unless your messages are encrypted and signed”
- Specify message encoding formats
  - HL7 v2 “pipes and carets”
  - HL7 v2 XML encoding
- HL7 v2 conformance profiles, v3 app roles

HL7 v2.x Conformance using Message Profiles

- Eliminates optionality
- Unfortunately message profiles are optional
- One-off standard for each system
- “V2 is so loose that conformance profiles don’t have the same meaning as for other standards” – Grahame Grieve, HL7 editor
- Not always expressive enough
  - How you use multiple patient identifiers
- Tools exist to help validate conformance

HL7 v3 Application Roles

- These are applied uniformly to all systems
- Roles are paired as sender/receiver
- Defines trigger events the system shall recognize
- Defines which messages are sent in response to trigger events
- Defines data content of the messages
- Vendor applications claim conformance to one or more application roles

Composability: a 5th SOA tenet?

- Services can be composed into larger systems and processes using workflow orchestration tools
- Good luck doing this with HL7 v2.x
  - V2.x message profiles are one-off mini-standards
  - Only ensures data compatibility, not necessarily full semantic interoperability
- This is better enabled with HL7 v3
  - Plug-and-play application roles
  - But more systems implement v2 than v3

Event Driven Architecture

- Further enables composite applications
- Buzzword bingo: “SOA 2.0” = SOA + EDA
- HL7 trigger events
  - Listening to ADT feeds = event subscription
  - HL7 v3 app roles formalize events even more

Pragmatic integration

- HL7 isn’t a silver bullet
- Do what works
- Be forgiving in what you accept, strict in what you produce
- Use conformance profiles where possible
- Think service-oriented, even with HL7