



# FHIR for Architects

## 架构设计人员FHIR教程

---

Lloyd McKenzie

2014年9月16日



# 关于本讲义



## ■ 下载地址：

➤ [http://gforge.hl7.org/svn/fhir/trunk/presentations/2014-09\\_Tutorials/FHIR for Architects.pptx](http://gforge.hl7.org/svn/fhir/trunk/presentations/2014-09_Tutorials/FHIR_for_Architects.pptx)

■ (use “anonymous” and email address)

## ■ 遵循Creative Commons使用许可协议，且具体为：

➤ [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/)



➤ 只要正确标明致谢信息，即可随意使用



# 自我介绍



- 姓名： Lloyd McKenzie
- 公司： Gordon Point Informatics (GPI)
- 背景：
  - FHIR三位主编之一
  - FHIR管理组共同组长
  - HL7建模与方法学工作组共同组长
  - HL7加拿大体系架构与基础结构工作组组长
  - 深度参与HL7和医疗保健信息交换工作15年(v2、v3、CDA等)



# 您呢？



- 您在HL7方面的背景如何呢？
  - v2? v3? CDA? 全新标准?
- 您扮演着什么样的角色呢？
  - 开发人员? 管理人员? 临床医生? 其他角色?
- 您要从今天课程当中获得的最重要的东西是什么？

# 教程目的



## ■ 您应当清楚：

- FHIR的互操作性范式（及其在何处运用）
- 适合将FHIR纳入体系架构堆栈的位置
- FHIR体系架构考虑事项及其如何处理
- 将概貌纳入体系架构解决方案的位置和方法
- 贵组织机构是否可能会、可能会何时及如何采用FHIR技术规范



# FHIR 知识回顾

# FHIR代表的是什么



- F – Fast – 快捷（设计和实施起来）
  - 只是相对而言）任何技术都不可能让集成的实现像我们所想的一样快
- H – Health/Healthcare – 健康/医疗保健
  - 这正是我们在此一聚的原因
- I – Interoperable – 互操作性/协同工作能力
  - 同上
- R – Resources – 资源
  - 构件 – 这些方面今后还有更多的新来者

# FHIR 宣言



- 关注焦点为实施者
- 对于常见场景的靶向支持
- 利用跨行业的网络技术
- 要求将人工可读性作为互操作性的基础水平
- 使内容自由可用
- 支持多种范式和体系架构
- 证实规范治理

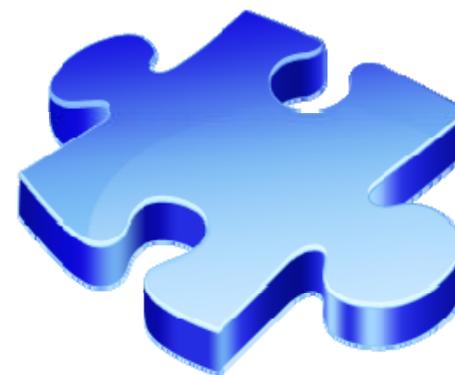


# 何谓资源 Resources



## ■ “资源”：

- 是逻辑上离散的小型交换单元
- 具有定义明确的行为和含义
- 具有已知的标识/位置
- 医疗保健领域所关心的最小交易单元
- 类似于V2的区段（Segment，段）和V3的公共消息元素类型（CMET）
- 三个组成部分：分立部分、叙述部分及扩展部分
- 共计100-150 种资源



```
<Patient xmlns="http://hl7.org/fhir">
  <extension>
    <url value="http://www.goodhealth.org/consent/trials"/>
    <valueCode value="renal"/>
  </extension>
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>Henry LEVIN the 7th, DOB 24-Sept 1932</p>
      <p>MRN: 123456</p>
    </div>
  </text>
  <active value="true"/>
  <identifier>
    <use value="usual"/>
    <label value="MRN"/>
    <system value="http://www.goodhealth.org/identifiers/mrn"/>
    <id value="123456"/>
  </identifier>
  <details>
    <name>
      <family value="Levin"/>
      <given value="Henry"/>
      <suffix value="The 7th"/>
    </name>
    <gender>
      <system value="http://www.hl7.org/v2/0001"/>
      <code value="M"/>
    </gender>
    <birthDate value="1932-09-24"/>
  </details>
  <provider>
    <type value="Organization"/>
    <url value="../organization/@1"/>
    <display value="Good Health Clinic"/>
  </provider>
</Patient>
```

扩展  
及指向其定义的引用

人工可读型摘要

- 标准数据内容:
- 病案号
  - 姓名
  - 性别
  - 出生日期
  - 服务提供者



---

# FHIR互操作性范式

## FHIR INTEROPERABILITY PARADIGMS

# 范式



- FHIR 支持 4 种互操作性范式



何时应当  
采用什么？

# REST

Representational state transfer, 表达状态传输



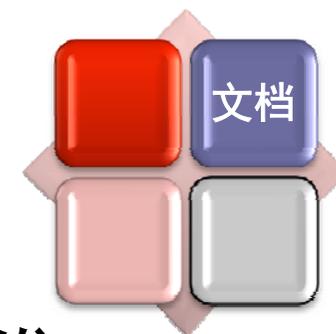
- 简单而又现成（开箱即用式）的互操作性
- 利用HTTP: GET、POST等
- 预先定义的操作
  - Create, Read、Update、Delete
  - 还有：History、Read Version、Search、Updates, Validate、Conformance 以及 Batch
- 作用发挥最好的环境：
  - 控制权位于客户端一侧，且存在着信任关系



# 文档 Documents



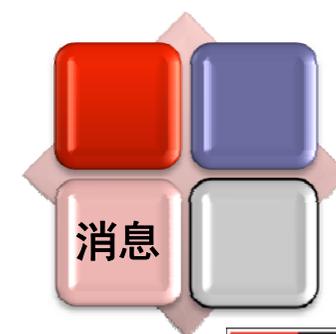
- 类似于 CDA
- 由若干捆绑在一起的资源所构成的集合
  - 根节点是一个组合式文档资源  
“Composition”
  - 就像 CDA 标头
- 以 ATOM feed（Atom 纲要）的形式发送
- 一种语境
- 对其可加以签字、验证身份等等



# 消息/报文 Messages



- 类似于v2和v3 消息传输
- 作为ATOM feed（Atom纲要），也是一种资源集合
- 允许针对**捆束（bundles）**的请求/响应行为
- 事件驱动型
  - 如发送检验申请，取回结果
- 可以是异步的



# 面向服务型架构

Service Oriented Architecture (SOA)



## ■ 随心所欲

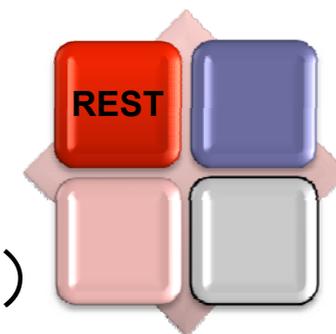
- (基于 SOA 原则)
- 超级复杂的工作流程
- 超级简单的工作流程
- 具体的资源或集合（采用Atom或其他格式）
- 采用HTTP或别的某种东西
- 仅有的约束限制就是，您是在采用某种形式或方式来传送FHIR资源



# 何时要采用REST?



- 希望系统之间为低耦合
  - 理论上，需要的提前协商极少
- 小型的轻量级交换
- 关注焦点是CRUD操作
  - 同样对于发布和订阅（publish/subscribe）
- 客户端驱动的客户端-服务器业务流程
- 服务器端点具有固定的位置
- 非常适合于移动应用、PHR及注册库



# 何时要避免采用REST?



- 复杂的或服务器驱动的业务流程
  - 操作次序很重要 (如复杂的决策支持)
- 工作单元 **!=** 资源
  - “Transaction” (交易)可能是一种选项
- 没有“**天然的服务器**”或者没有固定的网络位置
- 审计方面客户端缺乏信任等等问题



# 何时要采用文档？



- 关注焦点是持久性（persistence）
- 不涉及工作流程
  - 除了文档的提交/获取（post/retrieve）
- 对于已验证的内容，需要严格的规则
- 希望进行多种资源的通讯，且拥有关于如何呈现数据的控制权
- 数据涵盖/涉及多种资源



# 何时要避免采用文档?



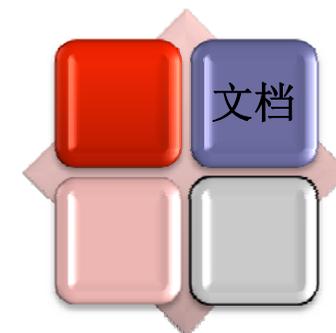
- 需要工作流程

- 请求/响应、决策支持

- 数据具有动态性

- 比如，希望查看现在的数据，而不是编纂时的
- 随着时间，存在多位贡献者

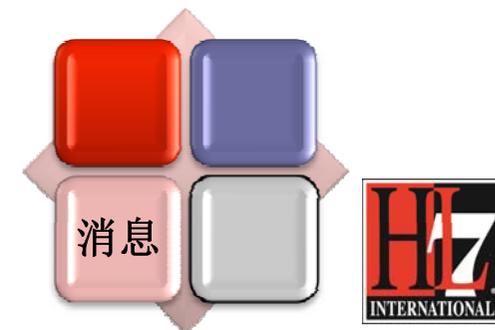
- 需要独立地对不同的资源加以访问/操作处理



# 何时要采用消息传输？



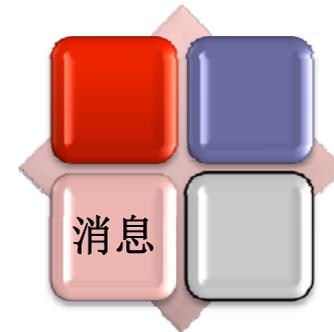
- 请求/响应工作流程
- 相对于针对单一资源的CRUD，需要驱动更为复杂的行为
  - 比如，合并、复杂查询
- 需要异步式通讯需要传输关于许多资源的信息但又希望交换最小化/尽可能最小
- 许多资源并没有“身份标识”



# 何时要避免采用消息传输？



- 对于如何实现数据的持久性或向人显示，需要加以精确的控制
- 需要轻量级通讯
- 希望避免关于行为的预先协商



# 何时要采用服务？



- 前三种范式都是服务接口的一种类型
- 当其他范式的能力无法满足需求时，则可采用自定义服务
  - 针对特定资源的非CRUD操作（如决策支持）
  - 比简单的请求/响应更为复杂的工作流程
  - 需要将文档持久性与行为混合起来



# 何时要避免采用服务？



- 当别的范式同样可以完成这项任务的时候
  - 也就是说，对于某种任务，如果本来早已就是由 REST、消息传输等等来处理的话，那就不要再定义自定义服务



# 关于范式的指导意见



## ■ 没有绝对的事情

- 请把上述关于告诫当作一种提醒
- 历史遗留者往往会推动方法，尤其是在初期
  - 比如，如果是v2后端，则可能会采用消息传输
- 体系架构的驱动因素将包括历史遗留需求、体系架构偏好、企业架构承诺

# 不同范式的组合使用



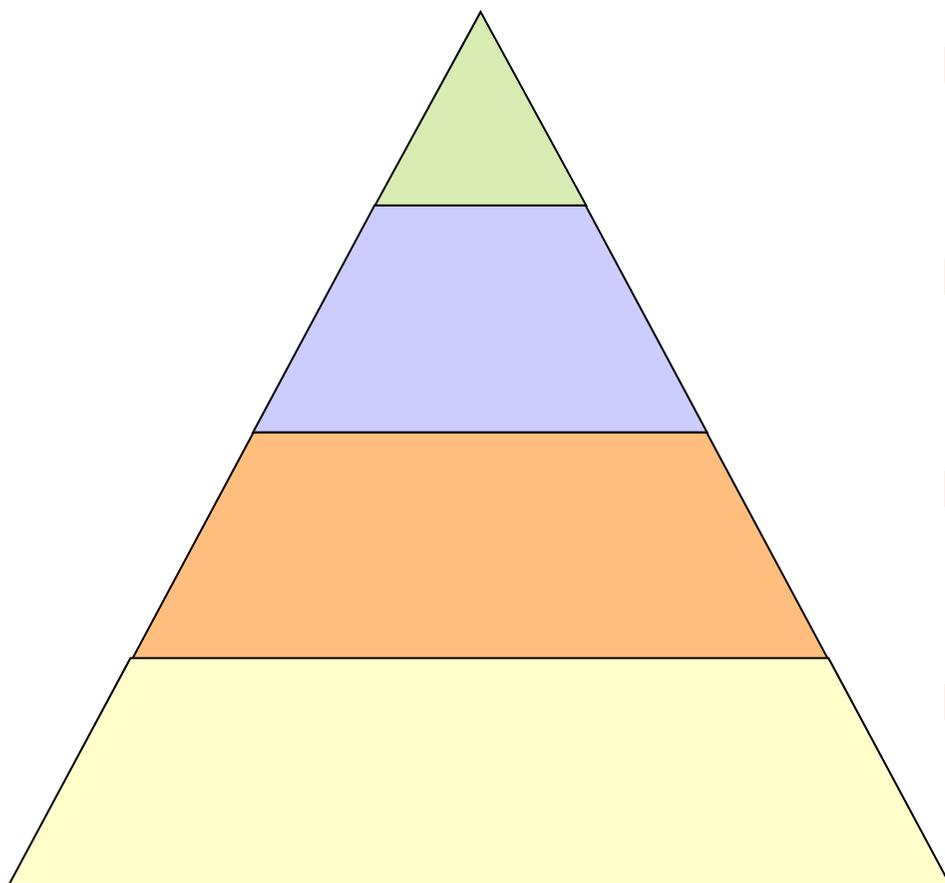
- 没有任何要求说一个系统只能支持一种范式
  - 比如，某家医院可能主要使用的是消息传输，但对于出院小结和各种报告采用的却是文档，在公开注册库和预约时则是采用REST，而为数不多的自定义服务则是用于决策支持或专用工作流程
- 跨越不同范式边界的数据共享（一般来说）较为容易

# 关于组合使用范式的告诫



- 如果入站的更新利用的是文档、消息或服务，仍需对RESTful版本标识符进行增量式更新
- 通常，应当整体持久存储文档，而不是利用组成部分进行重新构建
  - 旨在保证签名的有效性
- 遗留消息传输系统可能并不提供元数据，从而不能轻松地利用REST来公开或操作处理不同的资源

# FHIR 层面构成



- 业务流程
  - 如请求/响应
- 传输
  - 如HTTP、MLLP
- 范式
  - 如REST、服务
- 语法
  - 如XML、JSON

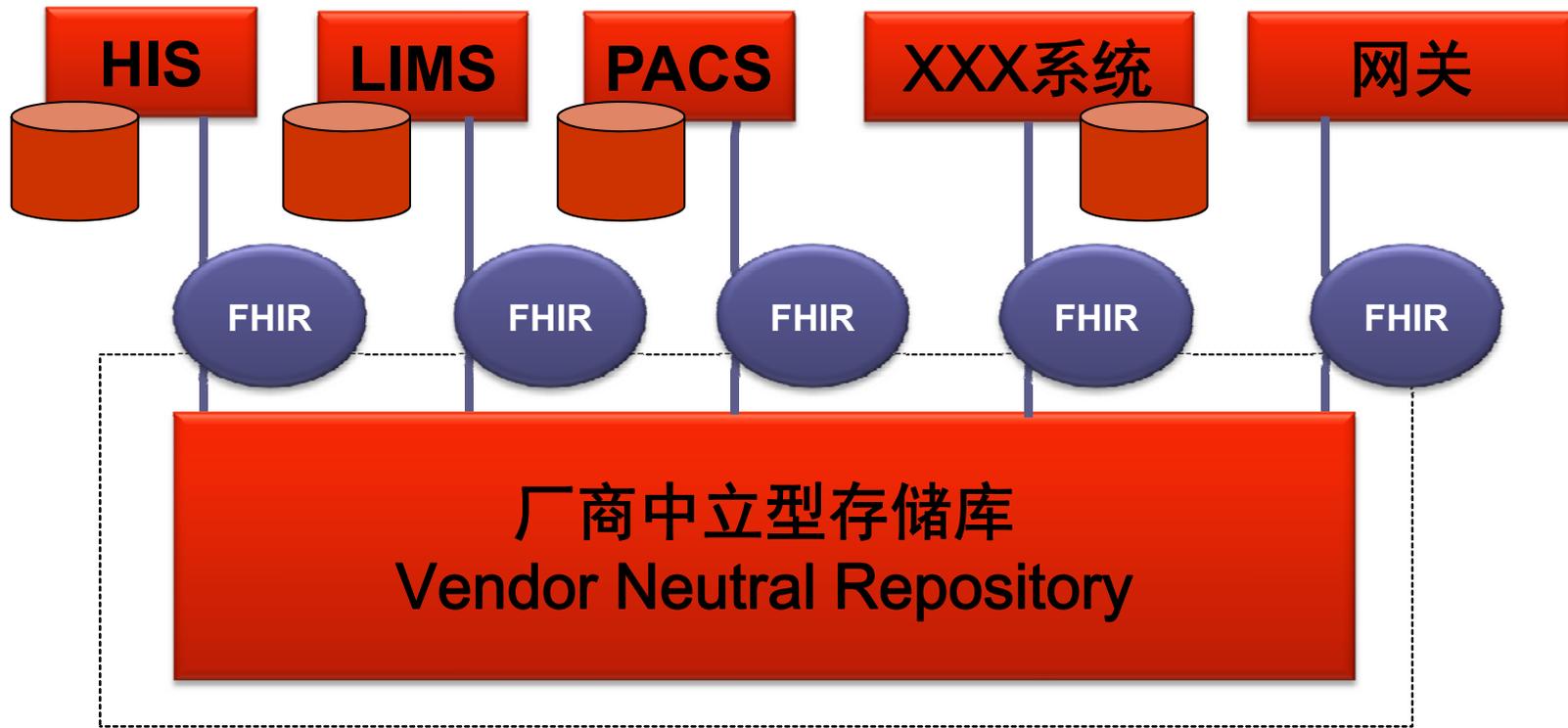


# FHIR 体系架构方法

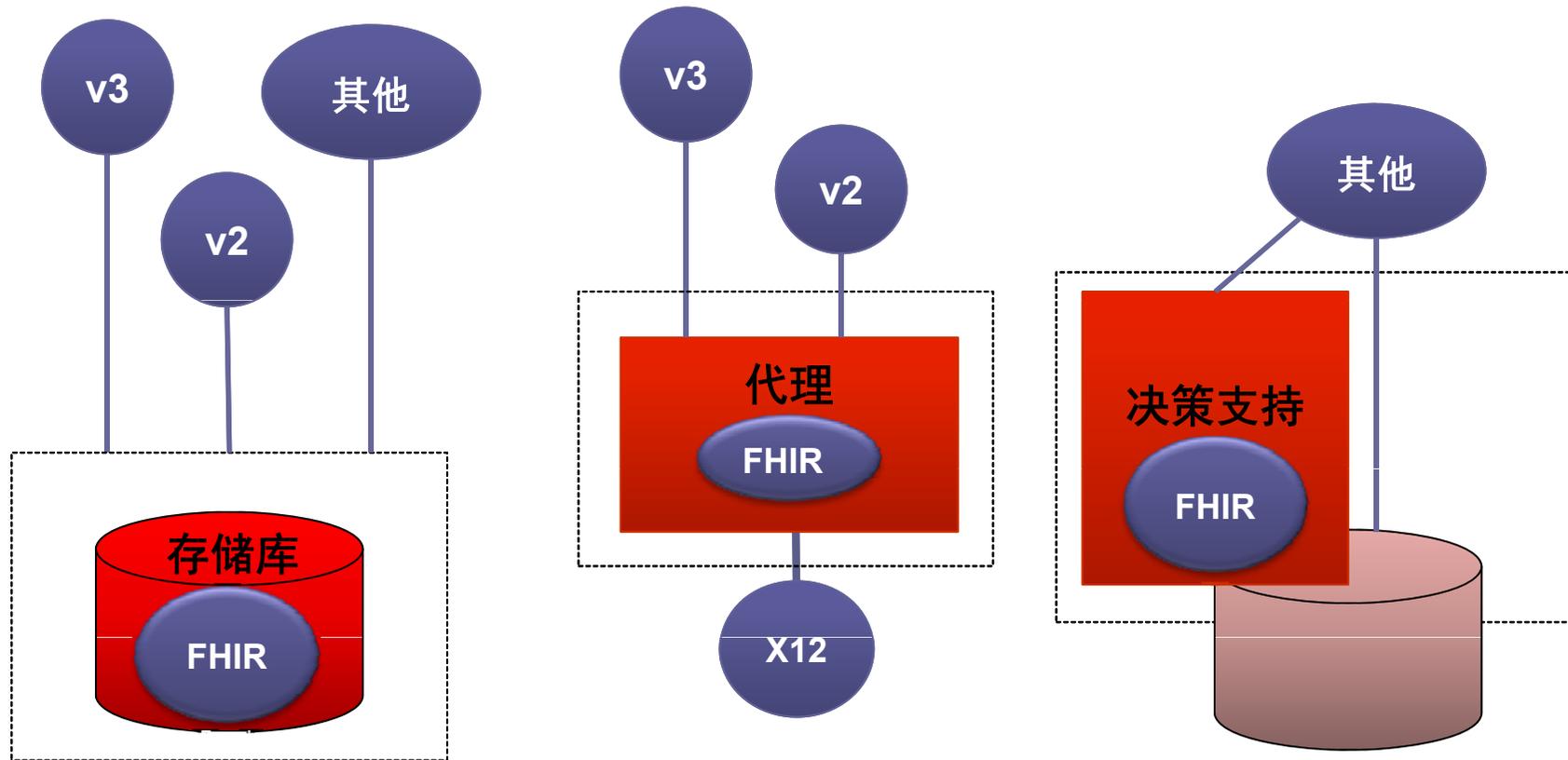
# 一些可能的情况



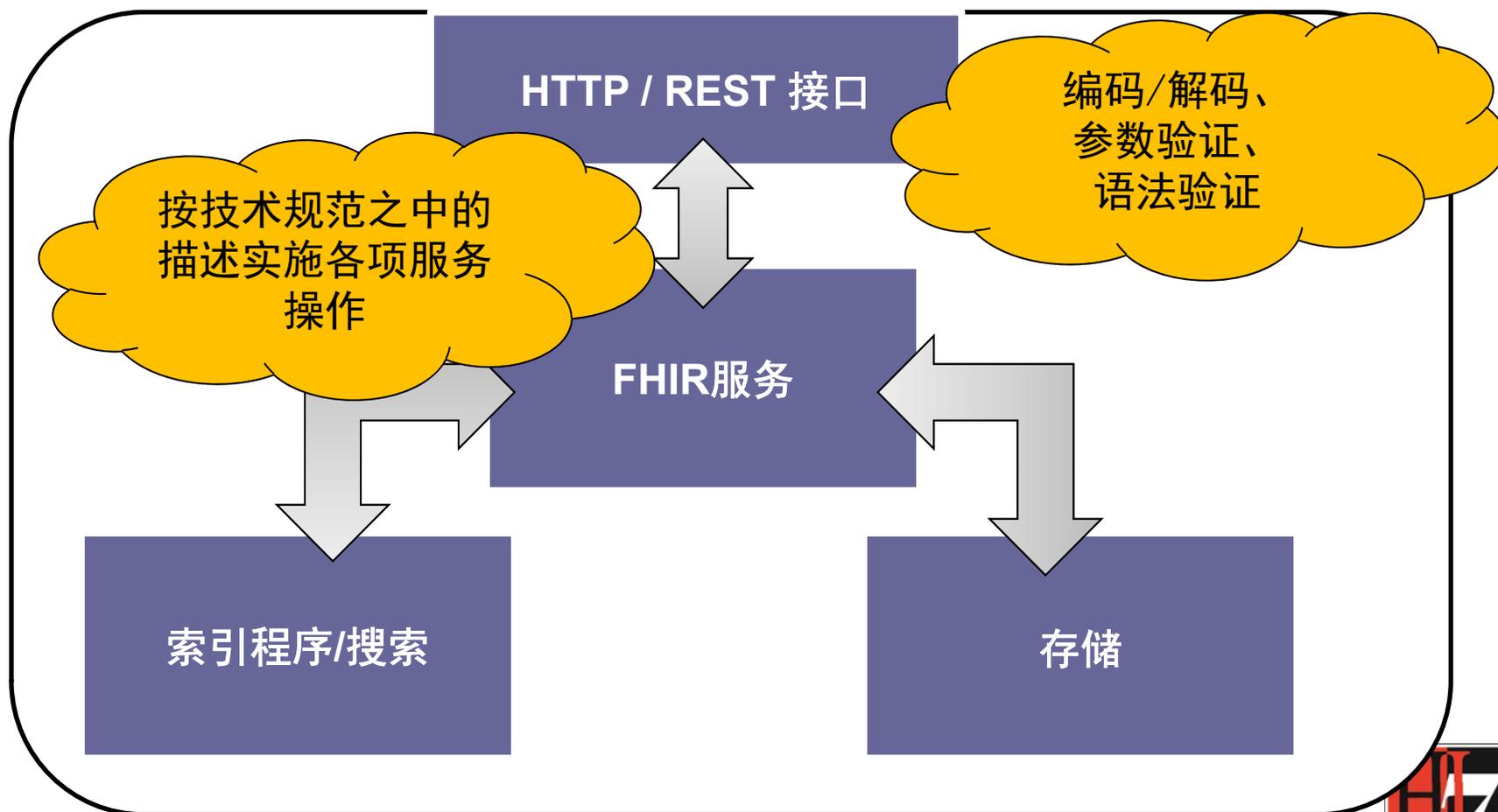
# 存储库模型



# 除了交换

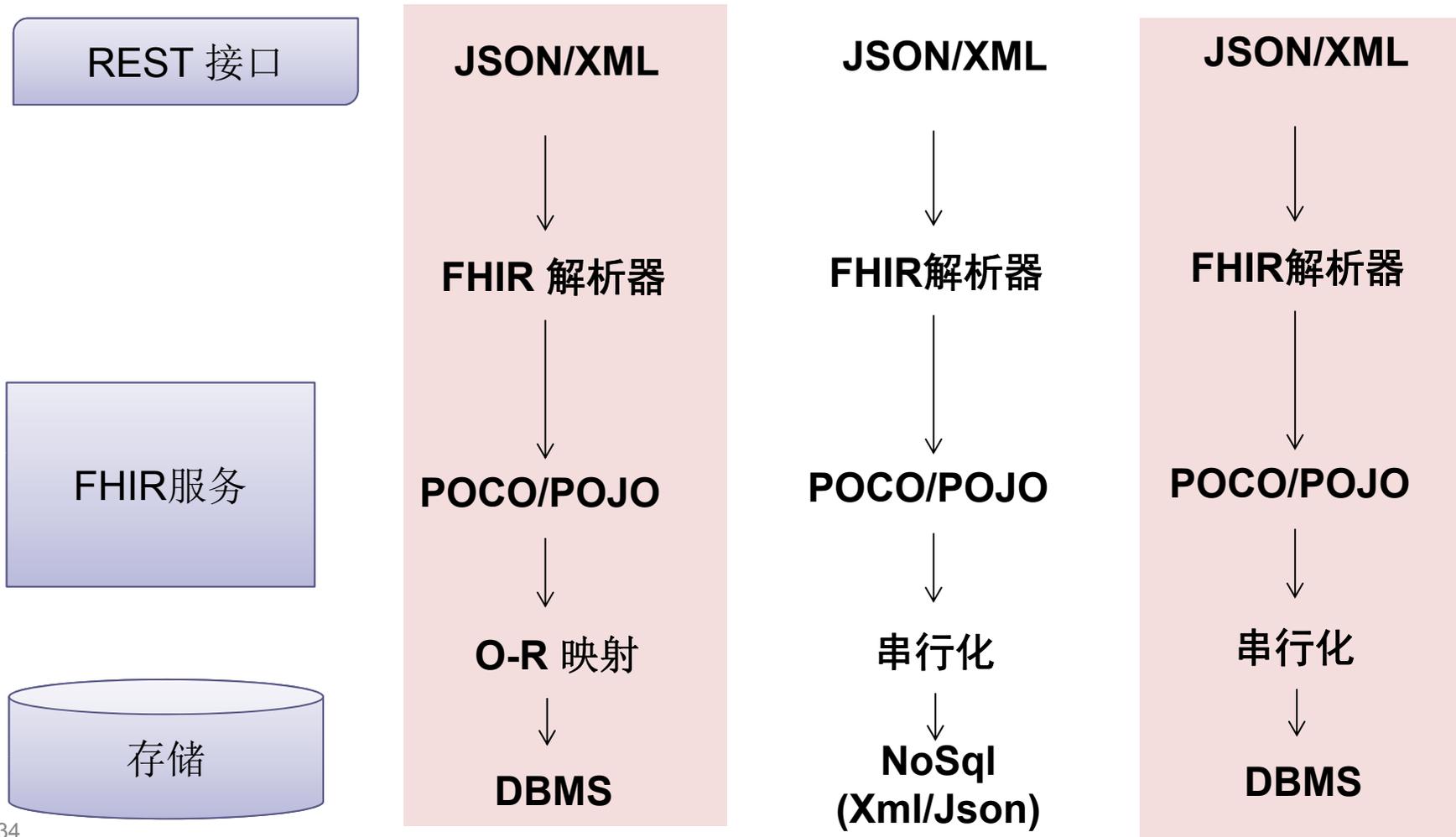


# 服务器概览



# 从有线格式到存储格式

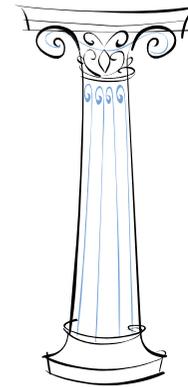
## From wire to store



# 体系架构



- FHIR对于系统的体系架构设计并未做任何假设
- 您可以将其用于：
  - 轻量级或重量级客户端
  - 中央式服务器或对等式共享
  - 推送或拉取
  - 查询或发布/订阅
  - 松耦合型或紧耦合型环境
  - 备有或没有历史追踪记录（版本）



# 底线



- FHIR是一套工具
  - 明确定义的各种资源
  - 可扩展性机制
  - 一套标准接口
- 主要目的是互操作性的数据交换
- 不过，可以采取多种方式来利用FHIR
  - 有很多连我们自己都还没有想到.....



---

以及那些与它们相一致的体系架构决策

# FHIR 的特点

# FHIR 的特点/特征



- 叙述 Narrative
- 扩展 Extensions
- 修饰型扩展 Modifier Extensions
- 版本 Versions
- 标签 Tags
- 语法 Syntaxes
- 签名 Signatures
- 参考库 Reference Libraries
- 元数据 Metadata
- 捆束 Bundles



# 叙述 Narrative



```
<DiagnosticReport xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <h3>CBC Report for Wile. E. COYOTE (MRN: 23453) issued 3-Mar 2011 11:45</h3>
      <pre>
        Test                Units          Value          Reference Range
        Haemoglobin          g/L           176            135 - 180
        Red Cell Count       x10*12/L     5.9            4.2 - 6.0
        Haematocrit          0.55+
        Mean Cell Volume     fL            99+            80 - 98
        Mean Cell Haemoglobin pg 36+           27 - 35
        Platelet Count       x10*9/L     444            150 - 450
        White Cell Count     x10*9/L     4.6            4.0 - 11.0
      </pre>
      <p>Acme Laboratory, Inc signed: Dr Pete Pathologist</p>
    </div>
  </text>
  <status value="final"/> <!-- all this report is final -->
  <issued value="2011-03-04T11:45:33+11:00"/>
</DiagnosticReport>
```

# 叙述 Narrative



- 除了一些局限的情况，预期所有的资源都会有叙述部分
  - 可能是自动生成的，或者是手工方式编辑的
- 决策
  - 应当是自动生成还是手工录入？
    - 自动生成型将简化接收方的处理工作
    - 有些文本将需要人工录入
    - 在某些情况下，可能会存在极少的离散数据



# 关于叙述部分的决策（续）



- 如果为自动生成，应当包括那些元素？又如何呈现它们呢？
  - 资源之中采取其他方式所编码的，且又需要人理解其中基本的临床和业务信息的所有内容
  - 一般都会包括修饰型扩展（modifierExtensions）
  - 可以包括其他的扩展
  - 最好请临床医生和其他人员审核其内容、呈现顺序及呈现形式
  - 要考虑到那些可能呈现在移动设备上的内容，因此，别把置标/标记搞得太花哨

# 关于叙述部分的决策（续）



- 是否应当向用户们显示叙述部分？
  - 利用叙述状态 *Narrative.status* 来决定
  - 无论如何，业务需求（如文档的证实/见证方面）都可能会推动关于呈现的需要

# 扩展



```
<Patient>
  <extension url="http://acme.org/fhir/Profile/main#trial-status" >
    <extension url="http://acme.org/fhir/Profile/main#trial-status-code" >
      <valueCode value="unsure" />
    </extension>
    <extension url="http://acme.org/fhir/Profile/main#trial-status-date" >
      <valueDate value="2009-03-14" />
    </extension>
    <extension url="http://acme.org/fhir/Profile/main#trial-status-who" >
      <valueResource>
        <reference value="Practitioner/example" />
      </valueResource>
    </extension>
  </extension>
  <!-- other data for patient -->
</Patient>
```

# 扩展



- 在FHIR之中，扩展乃是家常便饭的“常事儿”
  - 80%律的结果——让简单的东西保持简单
  - 扩展可以存在于任何地方
    - 的确如此，甚至是在布尔型或日期型内部
  - 符合性的系统不能因为实例之中含有未被识别的扩展而拒绝实例

# 关于扩展的决策



- 未被识别的扩展是否应当持久存在？怎么办？
  - 扔掉扩展 = 损失那些对下游系统可能有用的信息
    - 因此：请尽可能保留扩展内容
  - 可以采用二进制大对象或通用“槽位”结构来保存
  - 可以利用位置标签机制，在单独的二进制大对象之中保存特定资源的所有“未知”扩展
  - 有些历史遗留系统会无法处理这个问题
  - 当更新数据的时候，必须抛弃一些扩展

# 关于扩展的决策（续）



- 是否应当显示未被识别的扩展？
  - 扩展是采用URL来标识的，且具有某种已知的数据类型。可以解析其URL，查找其名称并加以显示
  - 成本/收益问题：有些扩展价值不大，而其他则可能很大
    - 可能会希望让用户去配置究竟要显示什么
  - 如果在进行查询，则可以考虑一下对扩展定义加以缓存，以便尽可能减少性能问题

# 关于扩展的决策（续）



- 作为扩展，该公开其什么？
  - 应当尽可能利用核心结构来公开数据
    - 仍可以在特定扩展之中发送相同的数据
    - 比如，采用或大或小的粒度、不同的编码、不同的数据类型
  - 在定义自己的扩展之前，先找找已有的扩展
  - 如果要使用扩展，在其应当通用，以有利于再用（继而获得广泛认可）
  - 扩展应当**位于**该扩展所描述的那个元素

# 关于扩展的决策（续）



## ■ 在何处注册扩展

### ➤ 考虑事项：

- 适用范围如何？对于本地扩展，选择本地注册库可能更有意义
- 是否有必要限制访问？
- 一般来说，更大范围内的注册 = 便于发现 = 更广泛的采纳 = 更广泛的认可

# 修饰型扩展

## Modifier Extensions



- 同样也是FHIR的核心组成部分
  - 需要此类扩展，因为无法安全地忽略某些扩展
  - 无法计算处理带有未加识别的修饰型扩展的元素。  
不过，可以：
    - 拒绝实例
    - 只是显示叙述部分
    - 获取定义并寻求人工审核

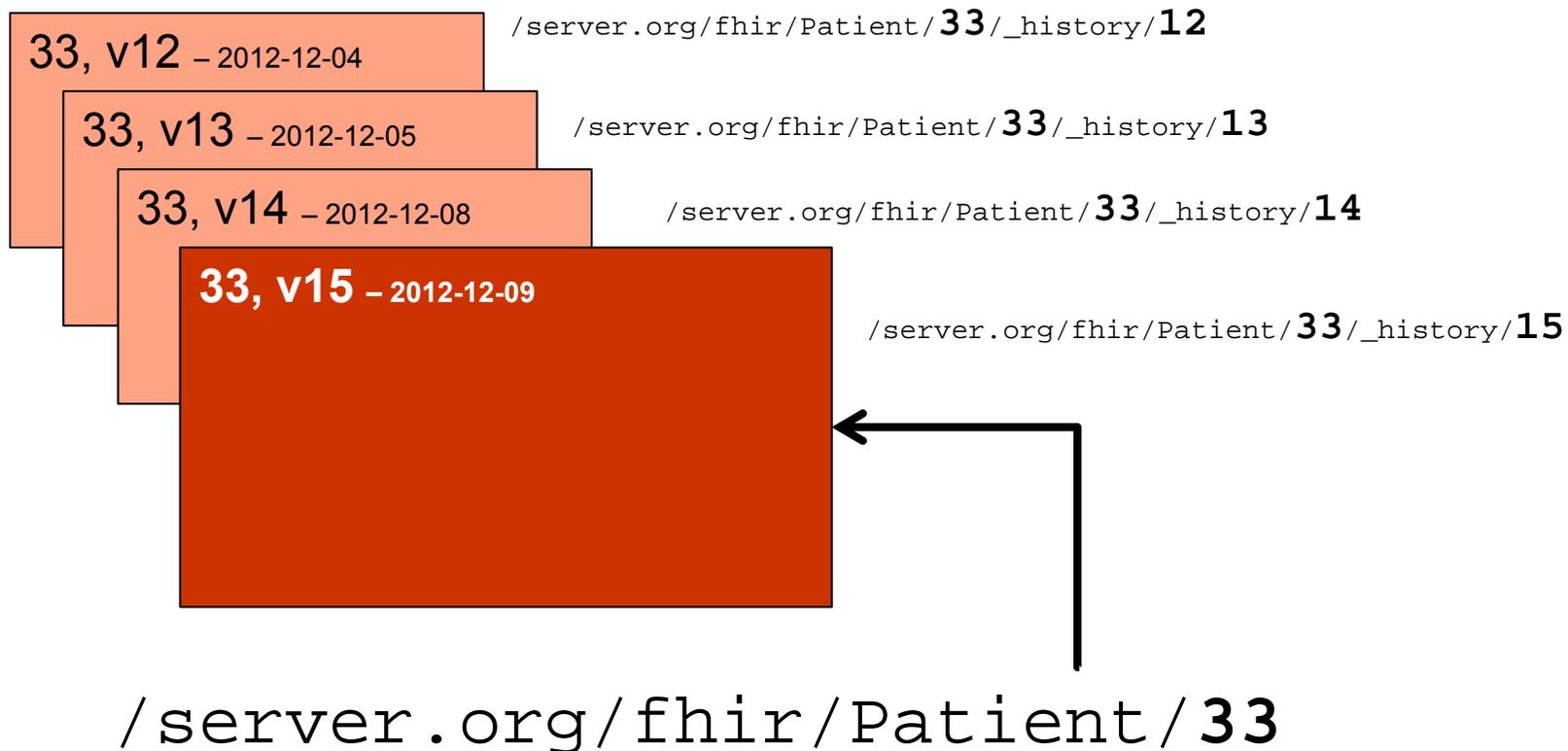
# 关于修饰型扩展的决策



## ■ 何时应当引入修饰型扩展？

- 修饰型扩展会破坏互操作性，因此：
  - 如果不用它也能达到目的，那就别用
- 考虑采用某种新的资源或是**其他资源** *Other*
- 是否能采用某种不会改变其他元素的解释的元素来满足需求吗？
- 当已经成为已有**实际工作**的组成部分，但因为所属方面过于局限而不能正当地成为技术规范组成部分的时候，最好采用修饰型扩展

# 版本 Versions



# 版本（续）



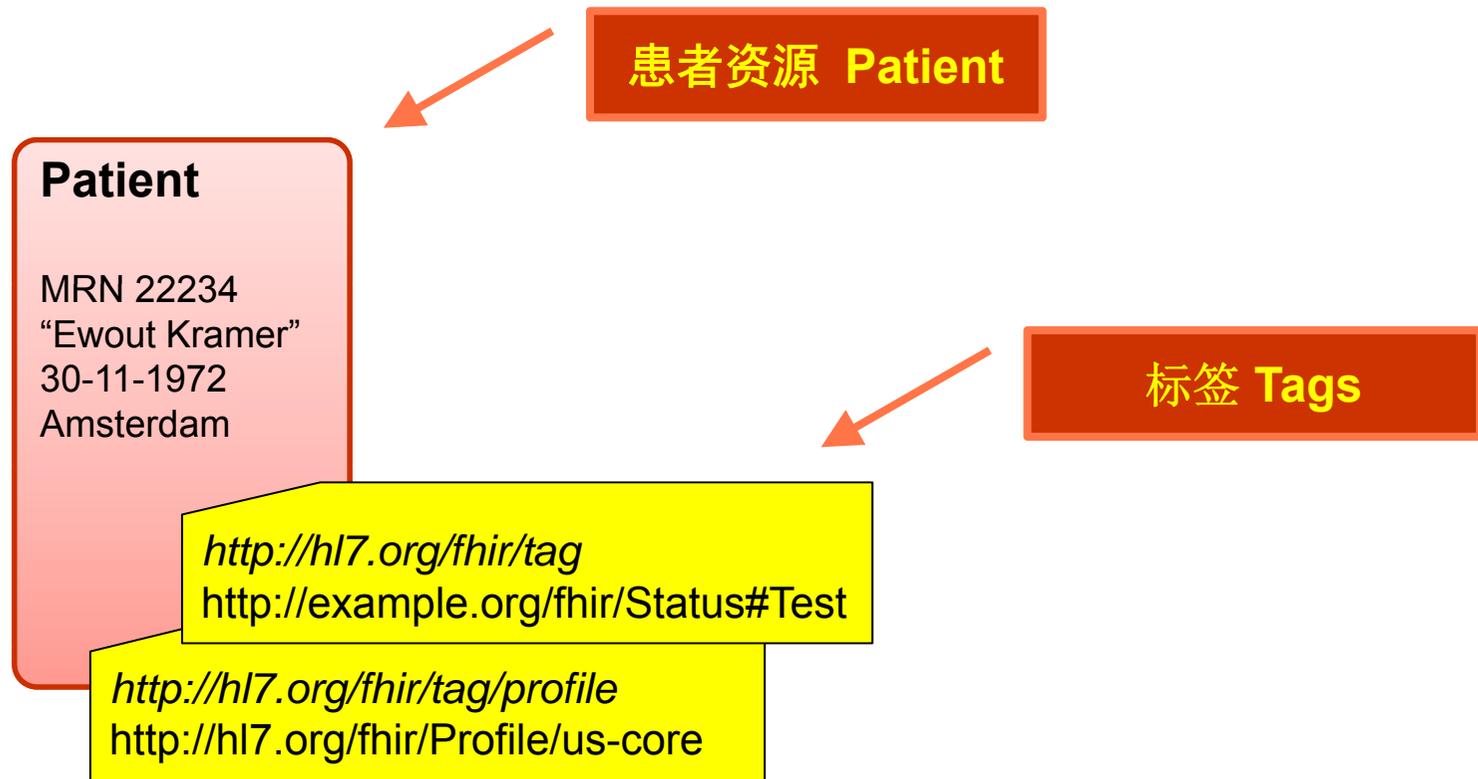
- FHIR允许跟踪记录和获取不同的版本
- 您希望支持版本控制吗？
  - 对于某些历史遗留的数据存储，可能并不容易或者是不可能
    - 仍将需要备有唯一性版本标识符（UUID、时间戳）
  - 提供实用的冲突检测机制

# 标签 Tags



- 允许将数据附加到当前资源“外部”的某个资源
  - 在添加/变更的时候，不要破坏签名
  - 可用于：
    - 分类（如安全性/访问控制、概貌）
    - 工作流程
  - 需要业务协议（业务方面的一致意见）
    - 假如编纂者不填写标签的话，就无法按标签进行搜索
      - 而且，还要按照相同的方式进行填写……

# 标签 (续)



# 关于标签的决策



## ■ 采用标签还是扩展？

➤ 如果是下列情况，请采用扩展：

- 与元素相关联的是业务对象，而不是电子记录
- 是资源经过证实的内容的组成部分
- 应当被包括在叙述部分当中
- 变更将会迫使出现新的版本

# 语法 Syntaxes



## XML

```
<XXX xmlns="urn:foo">
  <B a="c" />
  <C>甲</C>
  <C>乙</C>
  <D>甲</D>
  <E>
    <h:div>并不是
    <h:b>那么</h:b>
    容易</h:div>
  </E>
</XXX>
```

## JSON

```
{ "B": { "a": "c" },
  "C": [ "甲", "乙" ],
  "D" : "甲",
  "E" : "<div>并不是<b>那么</b>容易</div>"
}
```



# 语法（续）



- 多种表达形式
  - XML 或者JSON
    - 关于RDF的工作正在进行中
- 有若干的参考实施项目都同时支持上述二者及其之间的转换
  - 实现互操作性的最大化
  - 对于数字签名，相互转换方面还不足够稳健（空白分隔符就是个问题）

# 关于语法的决策



- 何时应当采用何种语法？
  - XML对于服务器来说必不可少
    - 正在重新审视
  - XML有着更多种类的手段和工具
    - XSLT、Schema、XPath
  - JSON占用带宽较少，且对于移动应用来说更加自然
  - RDF则属于小众/小生境/商机
  - 理想的情况是，服务器支持所有上述语法
    - 最大程度的互操作性

# 签名 Signatures



- 内容的数字签名有三种方式
  - 签署Atom 捆束（消息或文档）
  - 利用治理资源 *Provenance* 来签署资源版本
    - 受限于数据完整性
  - 扩展（适用于更为完整的签名）
- 并不要求对内容进行签名
  - 签名只是一种用于保证数据完整和/或不可抵赖性的机制
- 当在不同语法之间进行转换的时候，签名可能会保留不下来

# 参考库 Reference libraries



- 当前已有5个：
  - C#, Java, Pascal (Delphi), Objective C, Javascript
  - 还会有更多的吗？
- 可以完成解析、串行化、验证等等
- 还是**开源**服务器、基本客户端

# 关于参考库的决策



- 采用已有的参考库还是自己动手构建？
  - 与任何其他“构建还是复用”问题的评判标准相同
  - 自己构建的话，成本较高，但更便于调整
  - 至少，参阅一下参考实施项目，以便确保没有遗漏 FHIR 技术规范的细微之处
- 多久要更新一次？
  - 参考库变更频繁
  - 将需要对更新加以管控，尤其是在很可能将其作为自己代码的基础的情况下

# 元数据资源



- 符合性 *Conformance*、取值集合 *ValueSets*、命名空间 *Namespace*s、概念映射 *ConceptMaps* 及概貌 *Profiles*
  - 为系统的操作提供元数据
    - 考虑到“动态”配置
    - 比如，对于允许代码和必备元素，不是制定硬编码型的规则，而是去查阅取值集合或概貌资源
  - 可以托管在任何地方
    - 是否出于性能原因而需要一份本地副本？也许要通过发布/订阅来进行副本的同步化？



# 元数据资源（续）



- 常常按照层叠式层级结构来整理编排
  - 比如，国际级取值集合、国家级取值集合、本地级取值集合
  - 在该层级结构之中，需要考虑到用于传播上级变更的时间框架和机制

# 捆束 Bundles



- 适用于消息、查询、交易和查询响应的机制
- 理论上，可以利用不同的组成部分来重新生成文档，但是：
  - 关于究竟是不是文档组成部分的边界并不清楚
  - 无法保证条目的次序
  - 因此：建议将其存储为二进制形式

# 关于**捆束**的决策



## ■ 捆束之中的标识符

- 可以采用UUID、服务器标识符或版本专用标识符来标识捆束之中的资源
- 服务器标识符允许将文档之中的信息链接到现有的资源
- 如果数据对应于某个特定的版本（与筛选型视图相对），则应当采用版本标识符

# 关于**捆束**的决策（续）



- 资源究竟位于何处——捆束型、被包含型还是远程型？
  - 资源无法独立存在时，才应当采用被包含型
    - 父级资源被删除时无法存在
    - 没有足够多的信息可以解析
  - 如果某个资源应当遵循文档呈现规则，则必须将其包括在捆束之中
  - 如果某个资源是执行消息所请求动作所必需的，则应当将其包括在捆束之中，并不希望强迫接收方单独对其进行查询





# 其他考虑事项

# 其他考虑事项



- 标识的解析
- 数据的缺失
- 循环/回路（Looping）
- 可变的服务器能力
- 数据元的禁用
- 与历史遗留系统之间的互操作



# 标识的解析



- 资源是现实世界对象的电子表达形式
- 对于同一实际对象，可以有多个资源实例
  - 不同的服务器，或者有时甚至是同一服务器
  - 在不同服务器上，同一资源的标识符可能全然不同
  - 不同服务器上的数据可可能有所不同
- 一个资源多个地点（分别有着不同的标识符）
- 现成可用的数据可能会有所不同



# 标识的解析（续）



- 服务器内部及之间资源的匹配一般是利用业务标识符来完成的
- 还可以有业务“版本”
  - 关于业务版本的变更规则具有领域依赖性
  - 在可能维护既往版本的情况下，每个版本可能会分别是不同的实例

# 数据的缺失



- 必备型的资源或数据类型元素极少见  
( minOccurs > 0 )
  - 资源和数据类型具有语境非依赖性/语境独立性
  - 扩展可能会取代核心元素
- 因此
  - 不要假定数据就会存在
    - 始终要检查元素取值特征属性element/@value, 而不仅仅是元素本身
  - 为了强制实施相关的要求, 可能需要依据特定的概貌进行验证



# 循环/回路 Looping



- FHIR资源彼此链接成的是网络结构，而不是层级结构
  - 可能存在直接和间接的循环或者说回路
    - 在资源定义和实例之中如此
    - 即使核心元素之中不可能出现，扩展中也可能会出现
  - 解析和处理算法必须处理这种可能性

# 可变的服务器能力

## Variable Server capabilities



- FHIR定义了许多机制，用来为客户端提供查询控制能力
  - 分页、众多的筛选条件、\_include(包括)、\_summary(摘要)、compartments(区组/区划)
  - 不过，这些全都是可选的.....
- 那么，究竟服务器应当干什么呢？
  - 成本/效益的权衡
  - 您所支持的越多，与您合作的客户/客户端越多
  - 在某些架构下，有些能力的代价可能会非常高昂
  - 在合理的范围内，竭尽所能去做事情

# 可变的服务器能力（续）

## Variable Server capabilities



### ■ 客户端又该做什么？

- 利用所希望的那些能力，利用**有限的**一套服务器
  - 封闭环境下的效果较好
- 采用最低的那些能力，可以始于大多数/全部的环境
- 采用现成可用的高级功能特性，并在必要时将客户端处理作为备选退路
  - 客户端将更为高效但更为复杂

# 数据元的禁用

## Prohibiting data elements



- 在FHIR之中，不允许您禁止未知扩展
- 您可以将数据元设置为maxOccurs=0，不过：
  - 这会迫使客户端自定义究竟向您发送什么——不良做法
  - 更好的办法是接受并予以忽略
    - 在有限的情况下，如果忽略数据的话，可能造成责任（包括遭到起诉）问题，因而有必要加以拒绝

# 与历史遗留系统之间的互操作 Interoperating with legacy



- 如何让FHIR与v2、v3、CDA之间很好地配合使用？
  - 在此没有足够的时间涉及
  - 请参阅：
    - <http://hl7.org/fhir/comparison.html>
      - 公开
    - <https://healthlevelseven.desk.com>
      - 仅限于会员



---

# 概貌约束式 FHIR PROFILED FHIR

# 无概貌的FHIR Profile-less FHIR



- 要与FHIR互操作，并不需要概貌（profiles, 集成规范）
  - FHIR资源足够“离散分立”，足以明确地充填大多数的元素
- 方法
  - 充填/利用您所知道的所有元素，而对于额外的内容，则可采用HL7或国家的标准扩展
  - 尽可能多地与“所推荐的”术语集之间进行相互映射，并充填可编码概念型元素的文本 *CodeableConcept.text*
  - 公开符合性资源 *Conformance* 当中的那些能力



# 概貌的用途



- 不过，概貌依然相当有用
  - 可用于定义文档和消息
  - 可用于定义扩展、搜索参数
  - 设定特定语境下的互操作性期望
    - 不同的国家标准、照护服务类型、业务模式
  - 临床执业指南 (Clinical practice guidelines)/DCM (Detailed Clinical Model, 精细型临床模型)
- 仅仅接受那些您所能理解的（好处/坏处）
- 互操作性标准
- 临床执业指南 /精细型临床模型

# 利用概貌指导行为



- 概貌可用于动态地配置系统行为
  - 加载用于指导数据录入的概貌
    - 如肿瘤学转诊介绍信
  - 加载用于指导数据显示的概貌

# 同时应用多项概貌



- 可以针对一个实例同时运用多项概貌
  - 适用于不同取值集合的不同编码
  - 包括所有所需元素的并集
  - 在这些概貌并不约束最多出现次数时效果最佳

# 对于概貌的声明



## ■ 实例可以利用标签来标明其支持什么概貌

### ➤ 考虑事项：

- 这种声明是否具有版本特异性/具体版本所特有的？
- 您是否相信这种声明准确无误？
- 是否所有的客户端都会在提交时声明所感兴趣的那些概貌？
- 接收到数据之后才定义的所感兴趣的那些概貌怎么样？



---

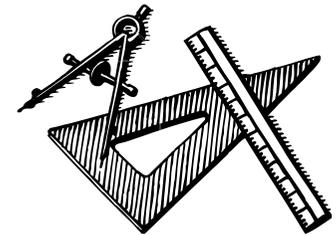
接下来做什么?  
WHAT NOW?

# 说说DSTU



## ■ FHIR目前还是一项“标准草案”

- 任何事情都可能发生变化——并不保证任何的兼容性
- 变更的驱动力来自于实施方面的反馈信息
  - 预期大多数变更将出现在资源里面
  - 利用参考实施项目和连通测试活动已经获得了相当多重要的实施经验
- 有些所需的资源尚未得到定义
  - 如保险、转诊介绍信等等
- 至少还会有一个DSTU/试行版，并且在正式成为规范标准之前可能会有更多的DSTU
  - 究竟什么及何时会变成规范标准，将取决于实施的程度

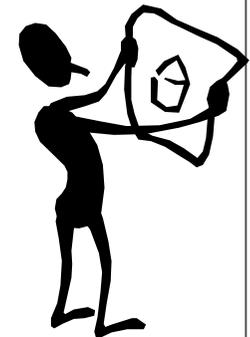


# 试行版策略

## DSTU Strategies



- 如果可能有多个版本的试行版同时在使用
  - 可以利用标签或不同的端点来加以区别
  - 准备好在版本之间进行转换，对元素进行移动或重新命名，或处理语法上的变更
- 对于缺少的资源
  - 请采用FHIR的[其他资源](#) *Other*
  - 创建您自己的自定义资源
    - 并不符合现有规范，但在试行期间且在封闭的社群范围内没关系

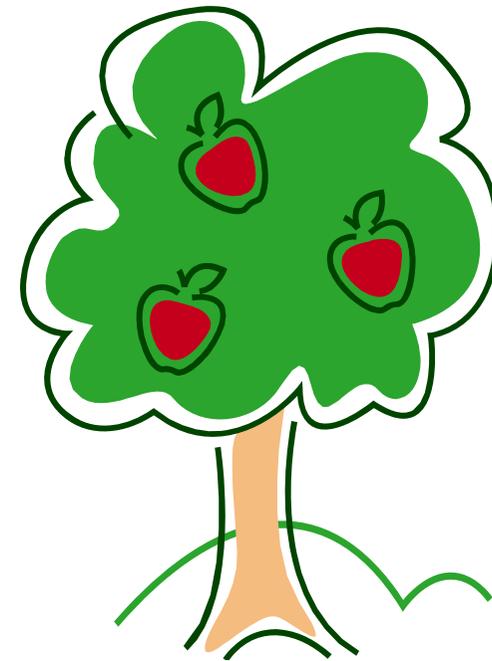


# FHIR 采纳方法



## ■ 低处的果子 (Low hanging fruit)

- 各种注册库
- 术语集
- MHD/健康文档的移动访问 (XDS)
- CCDA (联合式CDA)接口
- 患者门户 / 移动式健康应用
- 其他还有.....?



# FHIR 采纳方法（续）



## ■ 实验尝试

- 绿地（greenfield）：蛮荒之地
- 试点
- “很合身的”解决方案（移动设备、社会化媒体
- 在其他地方尚未标准化的元素
  - 调查问卷（Questionnaire）
  - 概念映射（ConceptMap）
  - 等等



# FHIR 采纳方法（续）



- 勇敢深潜
  - 勇者的“游戏”
  - 试行期间，可能还不成熟，因为FHIR技术规范很可能还会变更
  - 不必可少的是适应变化的能力



# FHIR 采纳方法（续）



## ■ 等待观望

- 等待下一个试行版、规范标准、上级指示（如美国的MU< meaningful use, 有效利用 >计划）
- 等待参考实施项目的稳定
- 拭目以待，期盼更多的实施经验

## ■ 不予理睬

- FHIR现在与我的境况无关并且也不可能相关

# 成本估计



## ■ FHIR解决方案的实施将会耗费多大的成本？

### ➤ 考虑事项

- 参考实施帮助
- 学习曲线较低
  - 如果不熟悉XML / JSON / REST，依然是条曲线
- “草率从事的”互操作性会速度更快
- 无法加快共识的形成
- 各种用于帮助映射内部代码和结构的工具，但仍要花时间
- 据说：相对于过去的互操作性标准，实施起来会“更快些”直至“明显更快地”



# 技能要求

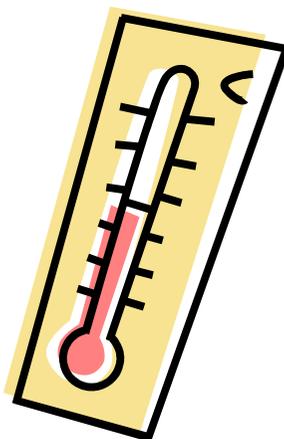


- FHIR 解决方案的实施工作需要您：
  - 具有关于XML 和/或 JSON方面的知识
  - 对HTTP（如采用REST）有一定程度的熟悉
  - 很可能还需要有个人精通HTTP安全性，或许还包括OAuth

# 考虑事项



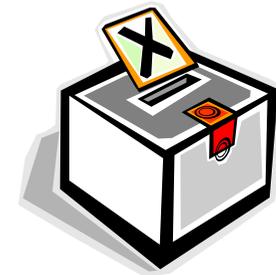
- 现在让您痛苦的是什么？
- FHIR能够如何解决这些痛处？
- 关于采用FHIR，您所担心的事情是什么？



# 重新评估的时间点



- 2014年12月：征求意见草案
  - 结构化数据采集
    - Structured Data Capture, SDC
  - 虚拟病历
    - Virtual Medical Record, VMR
- 2015年7/8月：发布下一试行版DSTU
- 2017年春季/夏季：首个规范发行版
- 每12~18个月发行一次进一步的版本



# 实施工作相关资源

## wiki.hl7.org/?title=FHIR



### FHIR Implementation

- The current specification: <http://www.HL7.org/fhir/>
- Help
  - [FHIR Starter](#) - aimed at the novice
  - [Help desk FAQs & knowledge-base articles](#) (members only)
  - [FHIR User Group](#)
  - [List server](#) - general questions
  - Skype conversation for implementers (**To Join** (david.hay25)) - for urgent enquiries.
  - [Stack Overflow](#) on-line application (use the tag fhir)
  - [FHIR gForge Tracker](#) for change requests/cc
  - [FHIR Tools Registry](#) - a list of useful tools for FHIR
  - [FHIR for Clinical Users](#) - an introduction to FHIR and how to migrate to the specification in the future
- Social Media on FHIR
  - FHIR blogs [David Hay](#) , [Ewout Kramer](#) , [Grahame Grieve](#)
  - FHIR News on Twitter [FHIR News](#)
  - FHIR Videos [HIMSS FHIR Session](#) , [HL7.tv](#) and [Ringholm](#)
- Testing
  - [Publicly Available FHIR Servers for testing](#)
  - [Open Source FHIR implementations](#)
  - [FHIR Connectathon 6](#) (May 2014, Phoenix)
  - [International FHIR Development Days](#) (Nov 24-26, Amsterdam)
  - [Historical Connectathons](#) (list)
  - [Organizations interested in FHIR](#)
  - [Profile Validation Tooling](#)



©

# 下一步您能做什么



- 参加其他的FHIR教程
  - 现有主题教程：架构师、开发人员、概貌
- 阅读FHIR技术规范：<http://hl7.org/fhir>
- 在维基上提出评论意见 (FHIR 技术规范上有链接)
- 在Twitter上跟踪#FHIR
- 参与塑造FHIR技术规范：
  - 参加此次工作组会议的FHIR分会场
  - 加入FHIR 邮件列表  
[http://wiki.hl7.org/index.php?title=FHIR\\_email\\_list\\_subscription\\_instructions](http://wiki.hl7.org/index.php?title=FHIR_email_list_subscription_instructions)
  - 试着去实施FHIR吧
  - 提出投票评论意见
  - 参加Connectathon连通测试活动



# 国际HL7 FHIR 开发者活动日 International HL7 FHIR Developer Days 阿姆斯特丹 2014年11月24-26日

## ■ 教育培训

- 共有14 个培训教程
  - 可供您挑选

## ■ Connectathon

- 与开发者同道会面
- 参与FHIR测试活动

## ■ 建立人际网络

- 随时可以联系到FHIR专家和编制人员



<http://fhir.furore.com/devdays>



**furore**

**Microsoft®**



# 提问环节



■ <http://hl7.org/fhir>

[lloyd@lmckenzie.com](mailto:lloyd@lmckenzie.com)

