架构原理

- 从架构的视角看Watson

艾飞, 高级软件架构师

大中华区软件集团 合作伙伴技术支持部

年轻时的我☺

爱互联网,爱自由,
爱历史,爱思考,爱技术,爱三俗,爱谁谁,
爱摄影,爱电影,更爱微博,
是IT民工,But和你們一樣还有夢想,
善良,真实,不装。

我不是成功人士，我是艾飞,
一个勤愤的架构师。我来自18M

Agenda

注：
部分技术内容由于公司的保密性要求已被删减

从IT薪酬的角度看位置

国家	软件工程师的年薪
美国	$57,381 - $83,633
巴西	$24,273 - $53,826
俄罗斯联邦	$11,466 - $30,190
中国	$8,998 - $24,781
印度	$5,622 - $9,885

Sources: Payscale, 2010 ; Ronin, 2007

IT行业早已进入“M型态”

选择做更有价值、更能为客户创造价值的人

Reference: 本页图片摘自互联网

架构师之路难行

1. 程序员老K：写了多年代码，年纪不小、压力很大、四顾茫然、举目无神，唉！要不要转架构师试试？

2. 架构师Z：架构师做了好几年了，依旧茫然，无人指点还要指导下面，搞得技术不像技术、销售不是销售。提升很难，瓶颈超大！

3. 老板某：花大价钱招了些所谓的“大师”和“牛人”，但好像对整个技术团队的能力提高不大，人一走茶就凉！持续性在哪里？难道只能靠钱砸人？

4. 客户Y处长：今天又来了几个架构师，讲的还不就那些东西，**忽悠**！接着忽悠！

5.架构师团队内部：同行相轻，知识传承难

难在哪儿

- **培养难**，贵，正规化、系统化难，无方法
- **提升难**，定位与选择更难
- **大师**常在，而优秀的**中坚层**难觅，如何提升团队战斗力
- **理想与现实** - 面对客户，要求高

  - 面对内部，岗位的局限性
- **挑战与未来** - 向传统IT企业学什么？

  - 向互联网企业学什么？

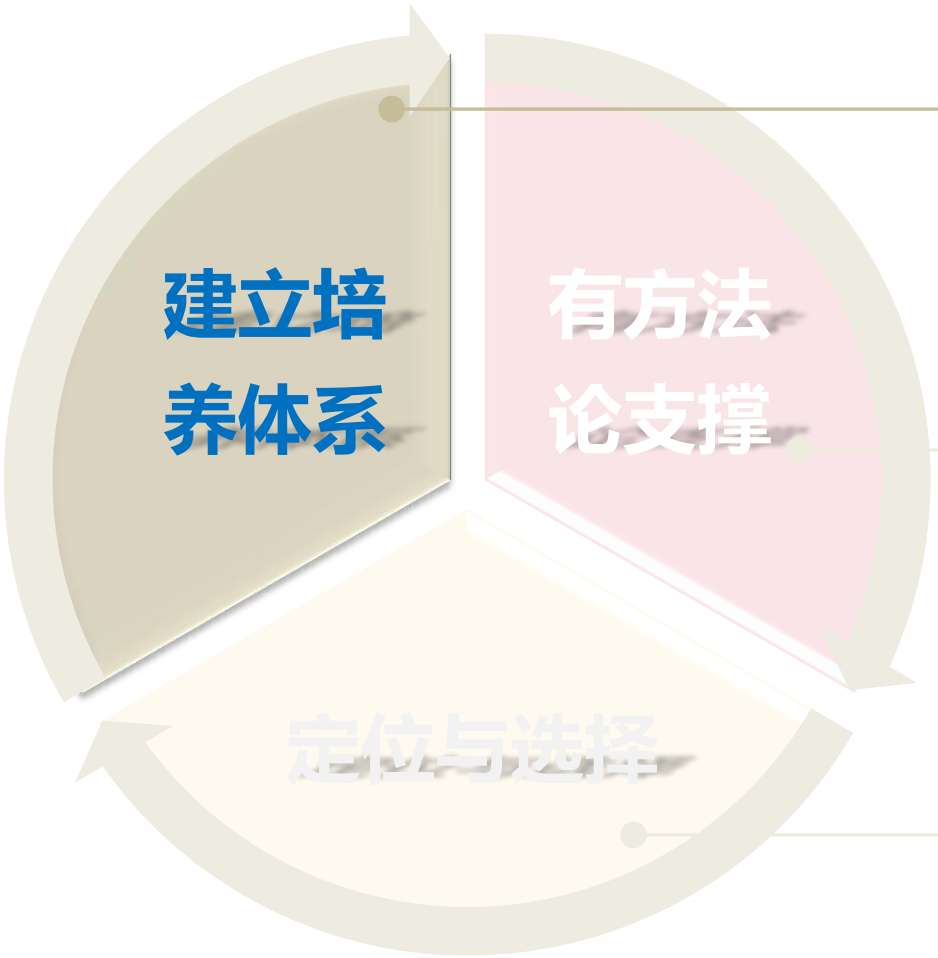
我们的尝试 - 自2010年7月起第一届BPITA训练营，与合作伙伴的架构师在实战同成长

- 架构师的职级构建
- 架构师的分类与认证流程

- 架构原理
- 系统化的方法论
- 行业知识

- 前瞻性
- 知识管理
- 拥抱开源

架构师的培养体系



- 架构师的职级构建
- 架构师的分类与认证流程

架构原理
系统化的方法论
行业知识

前瞻性
知识管理
拥抱开源

IBM架构专业的类别

Profession Structure

专业:

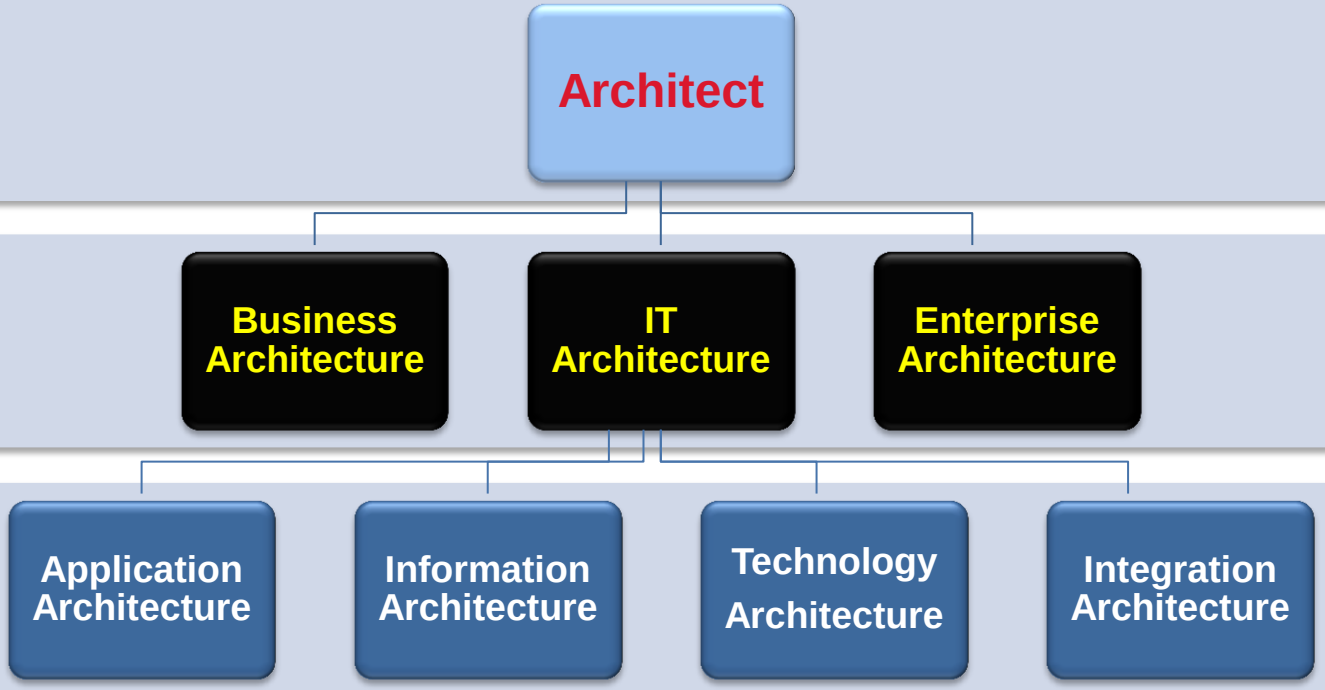
(IT Architect changed to Architect)

细分为3大专业域:

(Addition of Business Architecture and Enterprise Architecture)

4个子类:

(Consolidation from six to four sub-specializations)



IBM架构师的12种主要角色类别

1. Application Architect
2. Business Architect
3. Business Process Solution Architect
4. Enterprise Architect
5. Information Architect
6. Infrastructure Architect
7. Integration Architect
8. Operations Architect
9. Security Architect
10. System Engineering Professional
11. Technical Solution Architect
12. Test Architect

IBM架构师职业发展规划 - 看到前路，持续发展

IBM架构师职业模型

The Architect *demonstrates* these skills and capabilities in these levels

技术技能	核心能力	专业能力	领导力
Architect in Training	Associate Certified Architect	Certified Architect	Senior Certified Architect

ASSOCIATE CERTIFICATION | CERTIFICATION | SENIOR CERTIFICATION

At these levels the Architect *develops* these skills and capabilities

核心能力	专业能力	领导力	执行能力

The Development Checkpoints validate entry level skills for the next milestone

IBM Career*Smart* Framework alignment

初级	有经验的	专家级	领导级

检验级别的标准

Criteria Category	Foundation	Experienced	Expert	Thought Leader
Core Capabilities	18	18	18	13
Specialization Capabilities	N/A	N/A	5-8	Only when changing specialization
Experience Requirements	2	2	6	7
Experience Profiles	N/A	2	3	3
Professional Contributions	1	1	1 + mentoring 3 protégés	2 + mentoring 5 protégés
Education	N/A	Architectural Thinking, Method	Architectural Thinking, Method, Consulting, PM	N/A
Continued Education				Continuing development totaling 80 to 120 hours in past 3 years
Professional History	Yes	Yes	Yes	Yes
Letters of support / References		Yes		Yes

重认证级别的标准（每3年）

Criteria Category	Expert	Thought Leader
Core Capabilities	N/A	N/A
Specialization Capabilities	Required if changing specialization from last certification or recertification	Required if changing specialization from last certification or recertification
Experience	Activity Summary	Activity Summary
Experience Profiles	Optional 2 profiles in lieu of 3 references	Optional 2 profiles in lieu of 3 references
Professional Contributions	1 + mentoring 3 protégés	2 + mentoring 5 protégés
Education	N/A	N/A
Continued Education	Continuing development totaling 80 to 120 hours in past 3 years	Continuing development totaling 80 to 120 hours in past 3 years
Professional History	N/A	N/A
Letters of support / References	Optional 3 in lieu of experience profiles	Optional 3 in lieu of experience profiles

确认流程

Validation	Review Summary	Frequency
'*Foundation*' level validation	Manager + SME Review	On demand
'*Experienced*' level validation [Accreditation]	Manager Review + 3 SME Reviewers	On demand
'*Expert*' and '*Thought Leader*' level **re**validation [Recertification and Senior Recertification]	Manager Review + BU Executive + SME Reviewer(s) [1]	On demand (However it is typically clustered around the certification anniversaries)
'*Expert*' and '*Thought Leader*' level validation [Certification and Senior Certification]	Manager Review + BU Executive + Initial Package Review (IPR) + Board [3 interviews + consensus meeting]	2-7 Boards per year depending on the geography

[1] One SME Reviewer is assigned to review the package. If decision is to decline, two additional SME reviewers will review package to reach consensus.

核心能力映射到基础技能

Defining Architectures Capability Theme	Foundation	Experienced	Expert	Thought Leader	IT Architect Fundamental Skill
Architectural Methods	☑	☑	☑	N/A	Apply Methodologies
Architectural Modeling Techniques	☑	☑	☑	N/A	Use Modeling Techniques
Architectural Thinking (New)	☑	☑	☑	☑	
Architectural Asset Creation and Reuse (combined two fundamental skills)	☑	☑	☑	☑	Develop Project Output for Future Reuse & Use Existing Work Products
Architectural Decisions	☑	☑	☑	☑	Develop Client Requirements & Architectural Decisions
Architectural Development	☑	☑	☑	☑	Develop Solutions Architecture
Architectural Validation Strategy	☑	☑	☑	☑	Develop Test Strategies & Plans
Solution Assessment	☑	☑	☑	☑	Perform Technical Solution Assessments
Standards for Solution Creation	☑	☑	☑	N/A	Apply IT Standards in Creation of Solutions
Architectural Risk Management	☑	☑	☑	☑	Manage Architectural Elements of Project Plan
Client Relationship Management	☑	☑	☑	☑	Manage Client Relationships
Project Planning	☑	☑	☑	☑	Plan Projects
Stakeholder Requirements Management	☑	☑	☑	☑	Develop Client Requirements & Architectural Decisions
Architectural Leadership	☑	☑	☑	☑	Lead Individuals & Teams
Architectural Strategic Direction	☑	☑	☑	☑	Lead in Setting Technical Direction
Communication	☑	☑	☑	☑	Apply Communication Skills
Consulting Techniques	☑	☑	☑	N/A	Use Consulting Techniques
Negotiation	☑	☑	☑	N/A	Perform Negotiations
(Not in core capabilities, included in the Understanding Implementation Impact Experience Requirement)			☑		Lead Strategy / Design / Implementation of Solution
(Not included in the framework)					Architect Solution for Security

Key → New | Changed | Moved | Removed

架构原理与方法论

架构师的职级构建
架构师分类与认证流程

- 架构原理
- 合适的方法论
- 行业知识

前瞻性
知识管理
拥抱开源

架构原理——真的懂了麼？

架构

是构成一个系统的基础组织结构，包括系统的组件构成，组件间的相互关系、系统和其所在环境的关系、以及指导架构设计和演进的相关准则

系统

是一组组件集合，被组织起来完成一个或一组特定功能。包括了独立的应用程序、传统意义上的系统、子系统、产品线、产品系列、整个企业和其它利益相关方的组合

干系人

与系统有关的个人、团队或组织

愿景

架构决策

环境

决定了对系统的开发、运行、政策以及会对系统造成其它影响的环境和设置

使命

视图

视角

关注点

流程

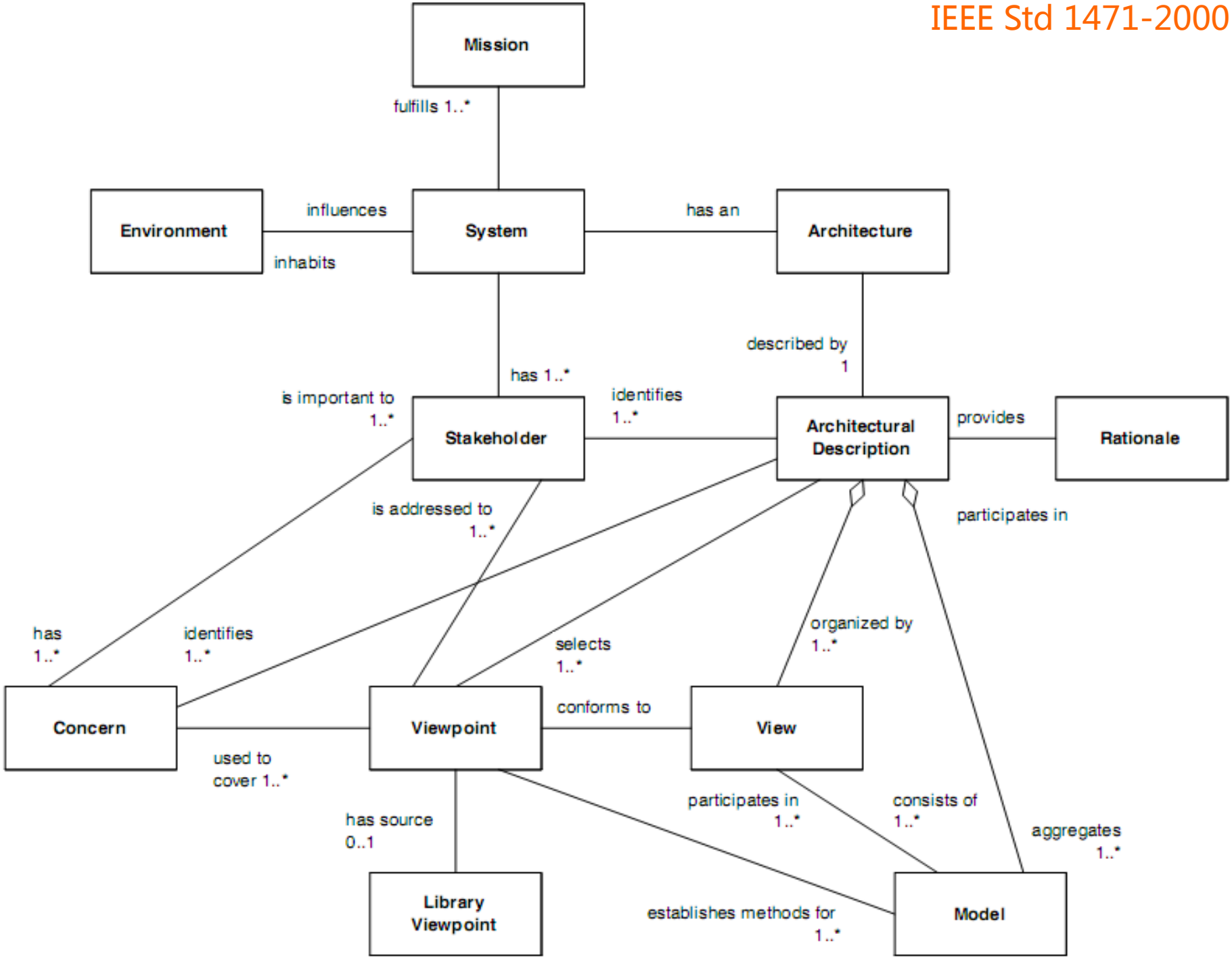
组件

子系统

模型

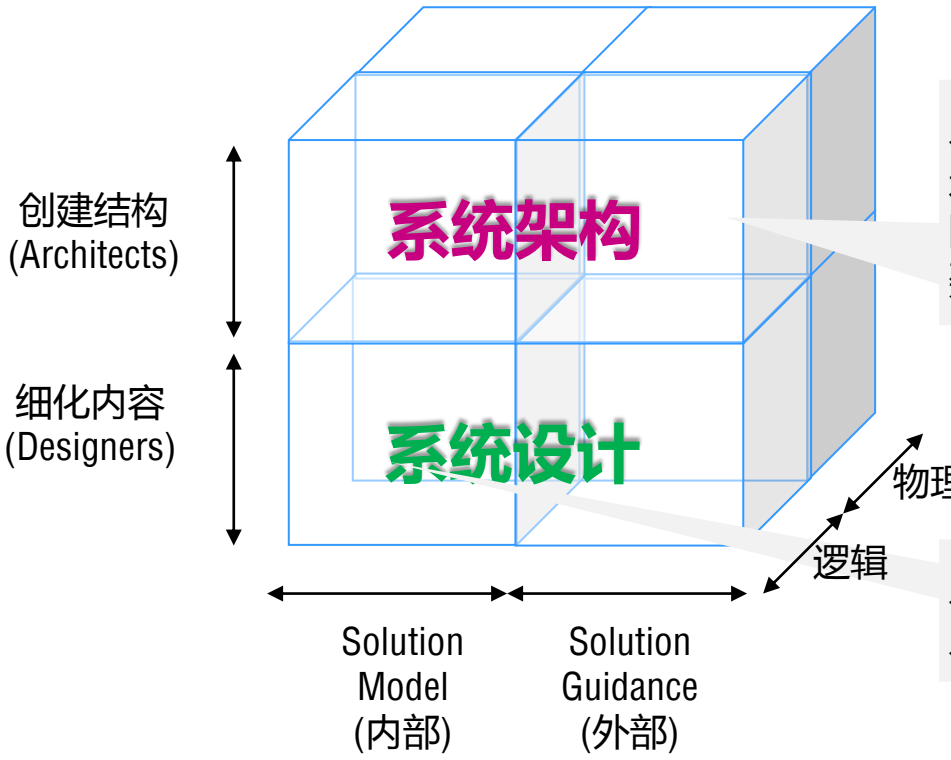
业务

IEEE Std 1471-2000



架构与设计

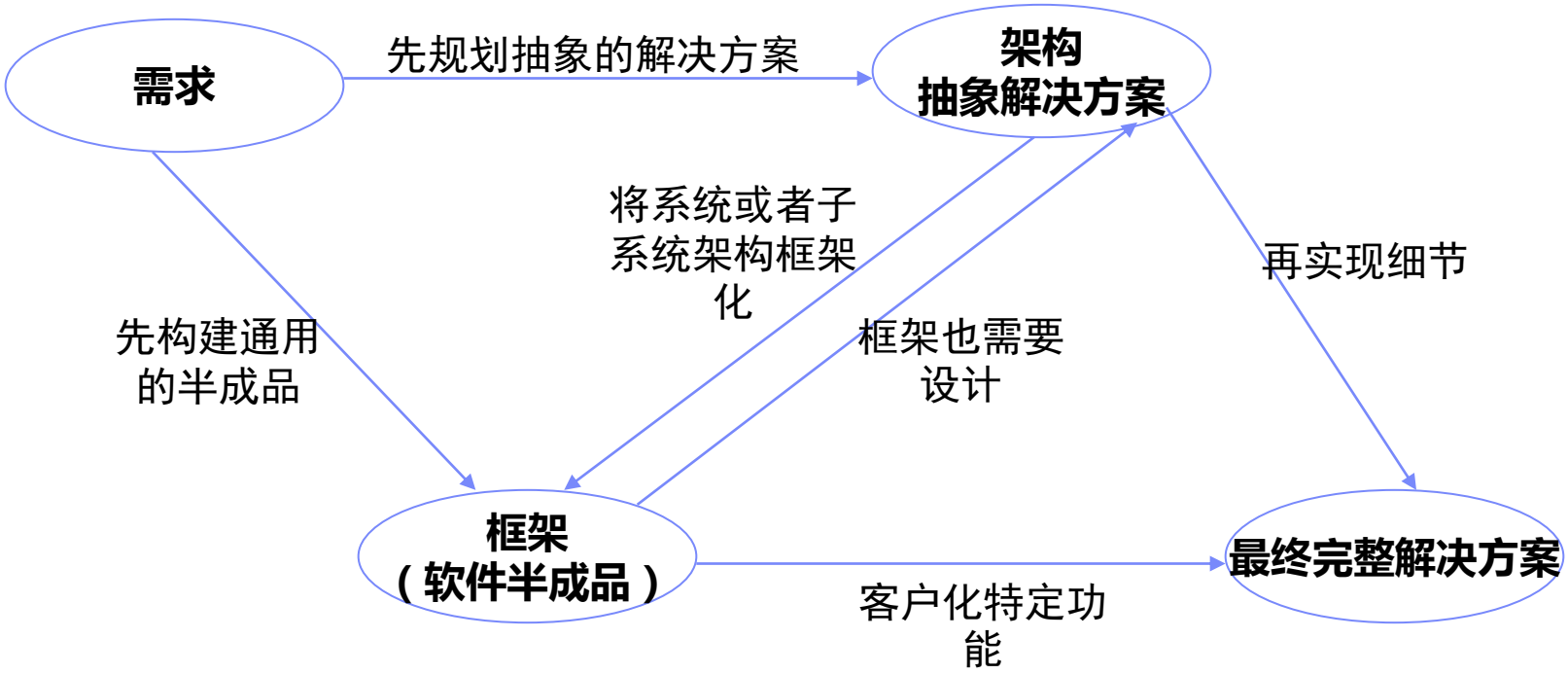
所有的架构都是设计，但不是所有的设计都是架构。架构代表了构建一个系统的重要设计决策，这里的“重要”程度是由变更成本的大小来衡量的[Grady Booch]



系统架构包括创建逻辑和物理层面，通常是技术独立的，且依赖于定义了解决方案的主要静态和动态元素的系统结构。系统架构决定了系统设计的上下文环境。

系统设计在系统架构的指导下，构建描述系统内部的行为和结构

框架是软件，架构不是软件

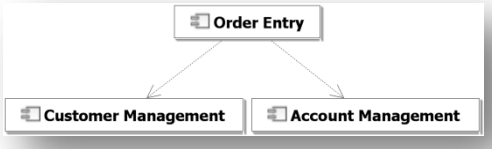


架构关注影响系统的重要元素

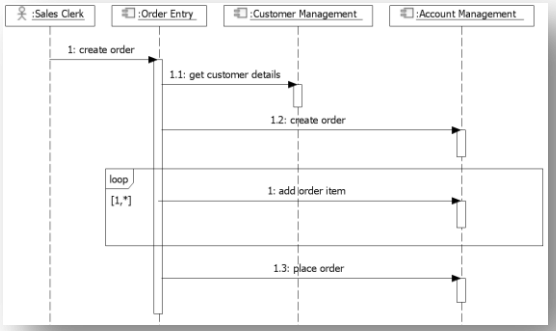
- **重要元素**
  - 系统的关键功能元素 - 如**事务处理**
  - 系统的重要特性 - 如**大数据的处理与存储**
  - 体系结构设计和实现中的难点 - **分布式计算，异构系统的集成**
  - 可能带来的技术风险 - **新技术(Hadoop商用)、新方法的使用**
  - 系统中状态不稳定的某部分 - **不可知的IDC稳定性**
  - 解决方案中的重要元素 - **认证机制**

架构的特性

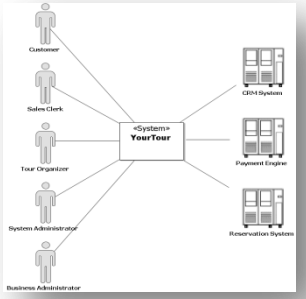
- 1　架构定义系统的结构
- 2　架构定义系统的行为和交互
- 3　架构只关注影响系统的重要元素
- 4　架构遵循一种架构风格
- 5　架构需要平衡干系人的需求
- 6　架构受所处环境的约束，反过来也影响它的环境
- 7　架构不仅仅要实现最后产出，还必须保证是合理和正确的
- 8　大多数的架构难点都和质量参数相关，而不是功能需求
- 9　合适就好
- 10　跟我重复：上下文、上下文、上下文



Structure

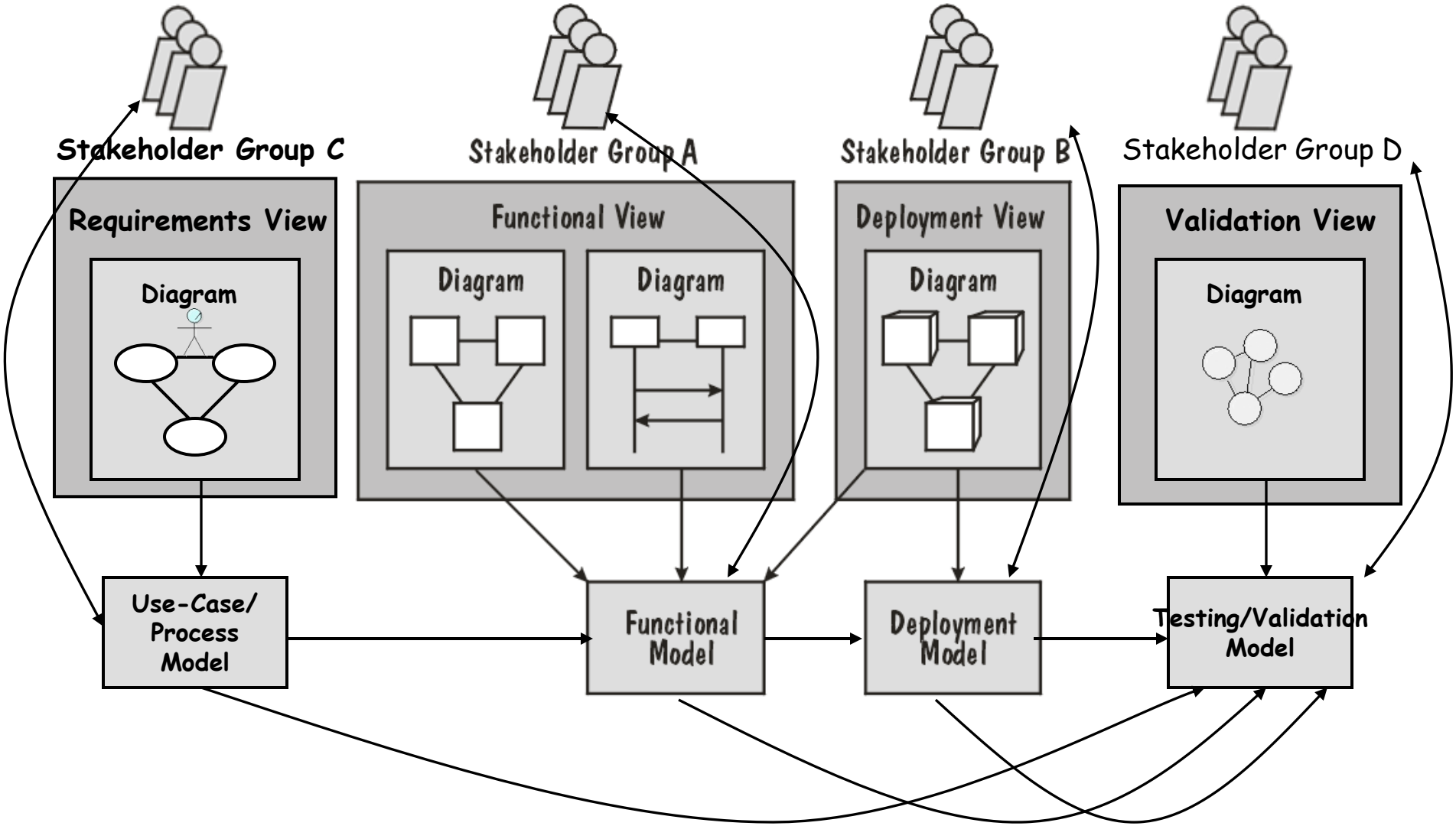


Behavior



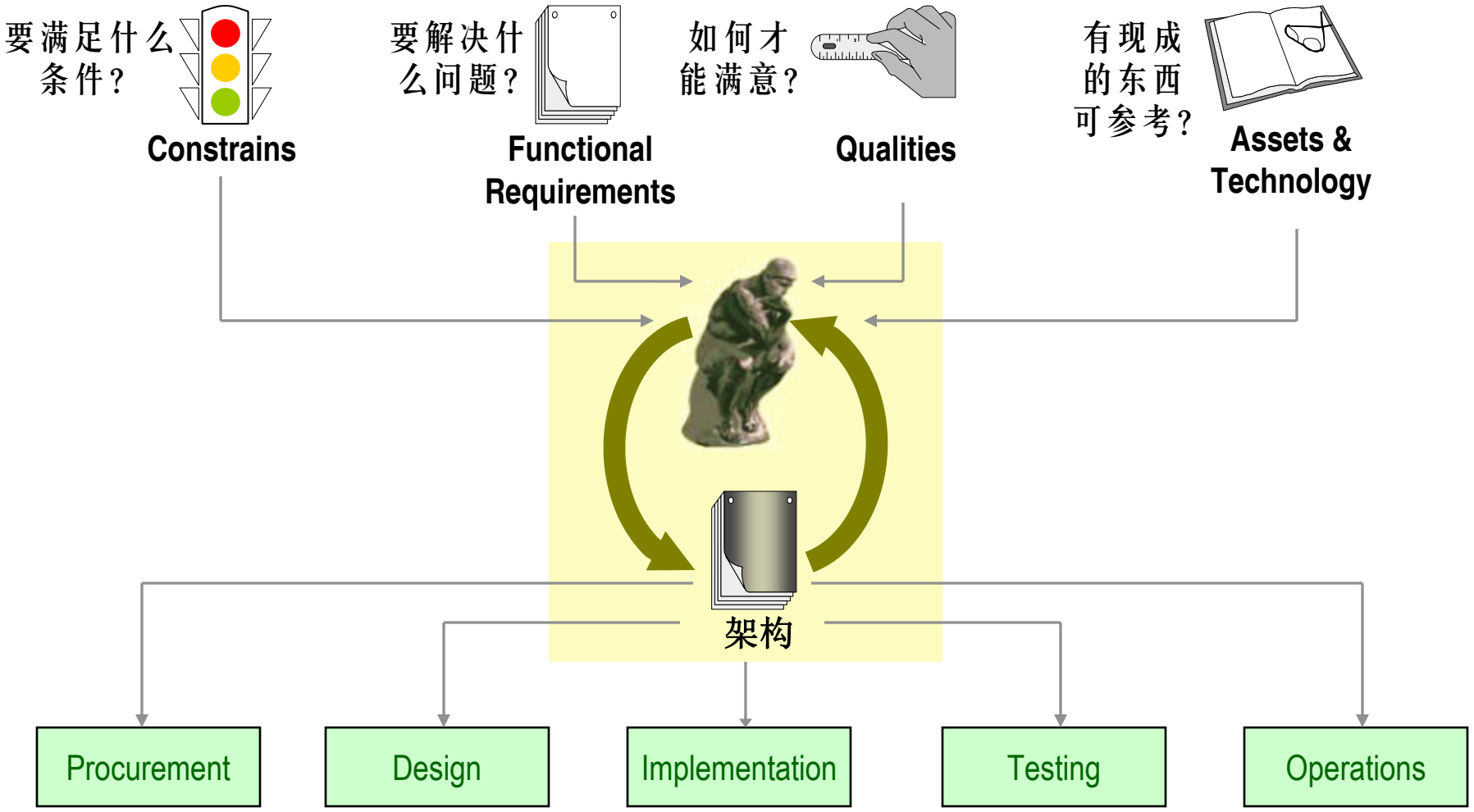
System Context

关注点→视角→视图→图表→模型→代码(产品/项目)

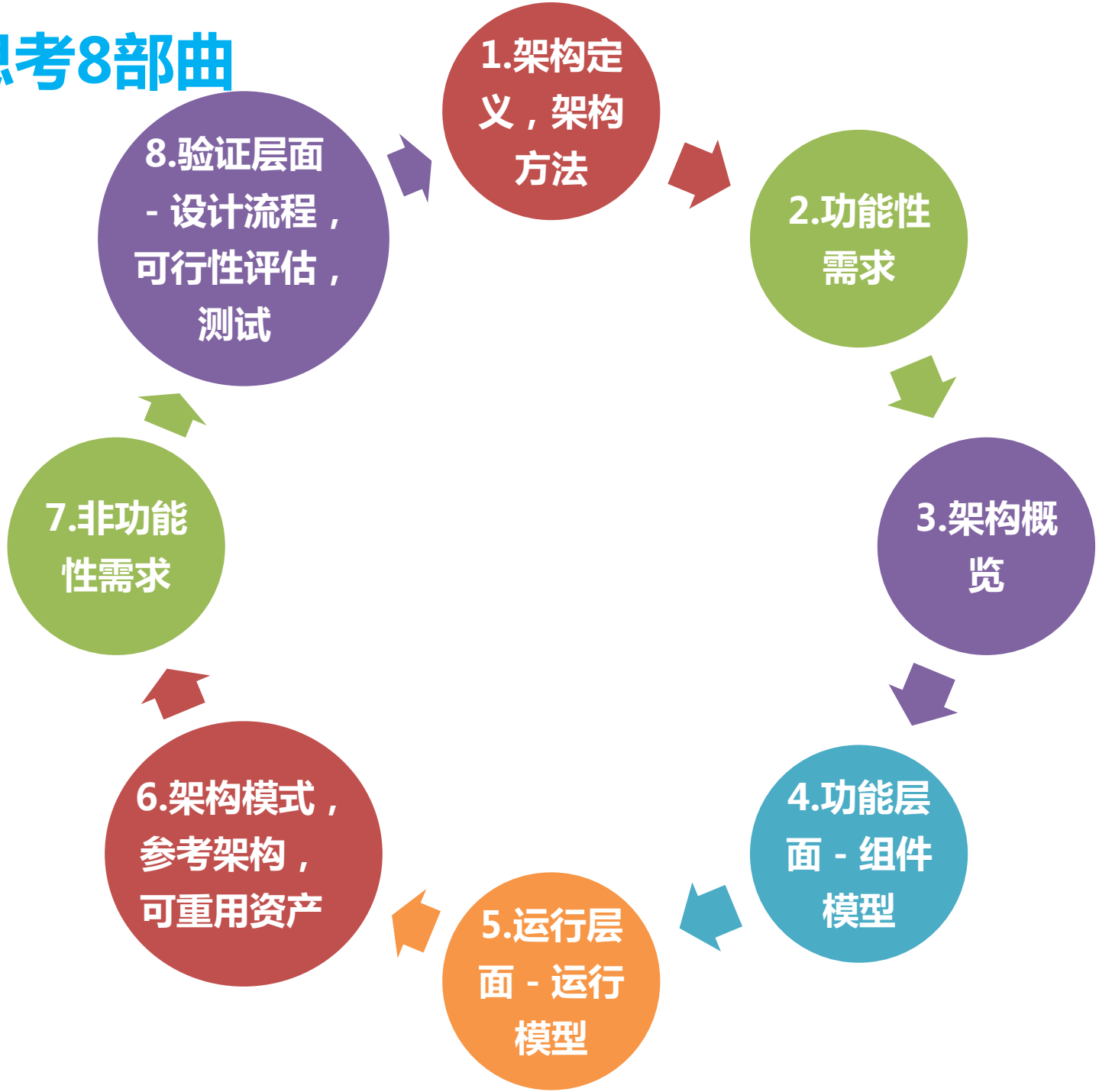


架构思考聚焦在一系列架构视角上

架构思考的过程


要满足什么条件？
Constrains
要解决什么问题？
Functional Requirements
如何才能满意？
Qualities
有现成的东西可参考？
Assets & Technology
架构
Procurement
Design
Implementation
Testing
Operations

架构思考8部曲


1.架构定义，架构方法
2.功能性需求
3.架构概览
4.功能层面 - 组件模型
5.运行层面 - 运行模型
6.架构模式，参考架构，可重用资产
7.非功能性需求
8.验证层面 - 设计流程，可行性评估，测试

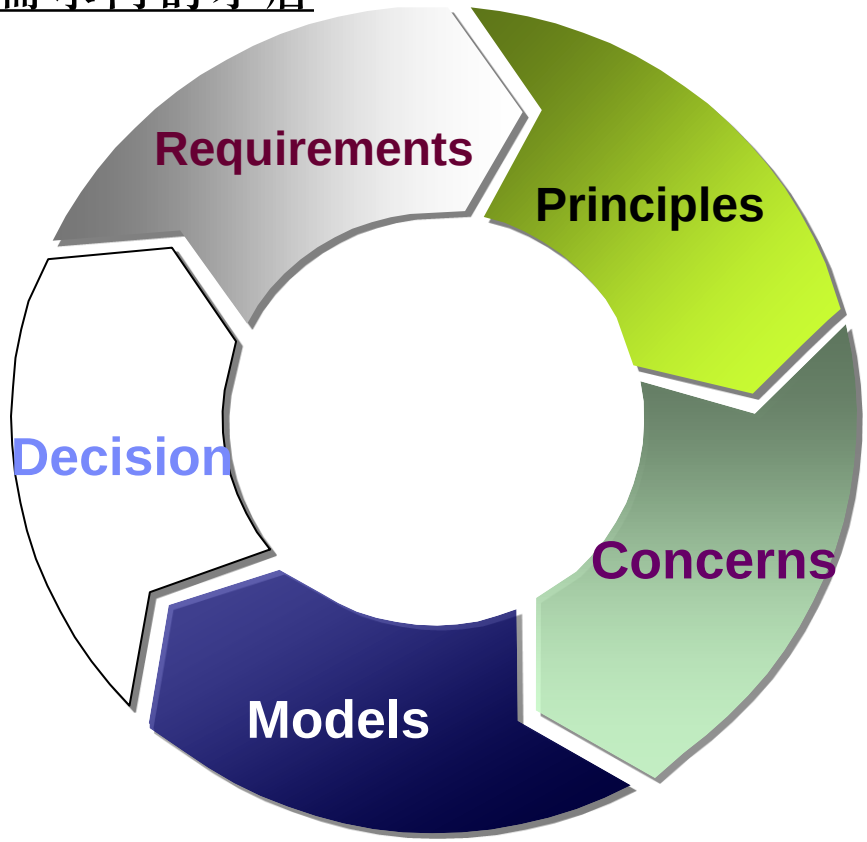
方法论

少走弯路，不丢失不遗漏，科学化、体系化，知识化

方法论涉及了各种方法和技术

•1分析需求，解决需求间的矛盾

•2多应用已验证的实践原则



•5提供决策路径

•3创建满足不同干系人关注点的视图

•4在正确的抽象级别上创建模型

方法论有助于达成一系列目标

1. 分解IT系统的复杂性
2. 分析功能性需求和非功能性需求
3. 规约系统的物理模型
4. 系统元素结构化和相互间连接的指导原则
5. 创建可控的决策路径，使系统随时间的变化有一个清晰的可溯路径和方向
6. 定义合成和分解规则

方法很重要，方法何其多

OMG OBJECT MANAGEMENT GROUP

FDA

PMI PMP

eclipse PROCESS FRAMEWORK

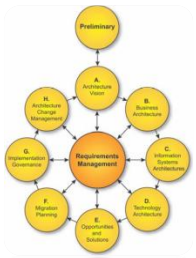
SPEM 2.0

Eclipse Process Framework

UML

DEPARTMENT OF DEFENSE UNITED STATES OF AMERICA

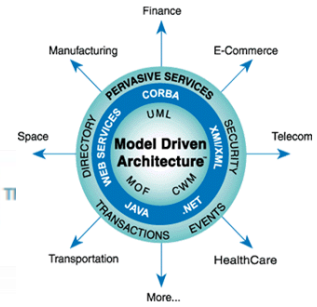
IBM Global Services Method



Unified Method Architecture

ITIL

TOGAF



OMG Model Driven Architecture

RUP Rational Unified Process

Six Sigma

CMMI

OpenUP

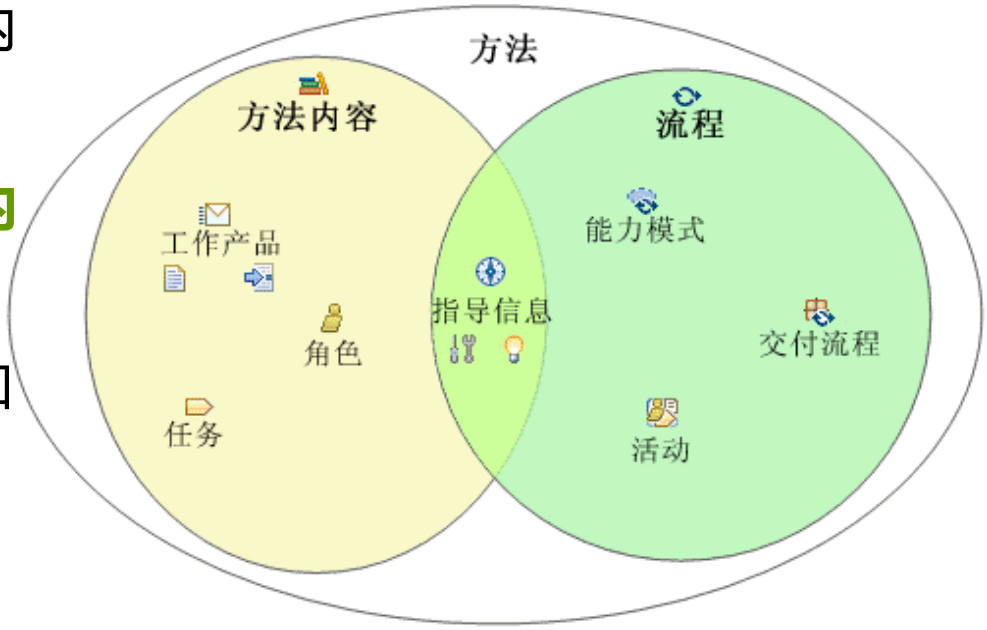
PRINCE2™

SCRUM

ISO International Organization for Standardization

方法的方法 - UMA方法元模型

- UMA元模型的目的是统一不同方法和流程的工程语言，用于表示由方法内容和流程组成的方法的模式和术语。
- 最基本的原理是将**可重用核心方法内容**与其在**流程**中的应用**分离**。
- 提供了IBM内部统一的方法论框架和术语定义
- 2007年提交至OMG成为SoftwareProcessEngineeringMetaModel(SPEM)2.0标准



1. 方法内容描述不受生命周期约束的元素。如角色、任务和工作产品
2. 然后流程取得这些元素，并定义一个应用它们的顺序。

架构在不同的阶段都有体现

TECHNICAL ARCHITECTURE METHOD PHASES

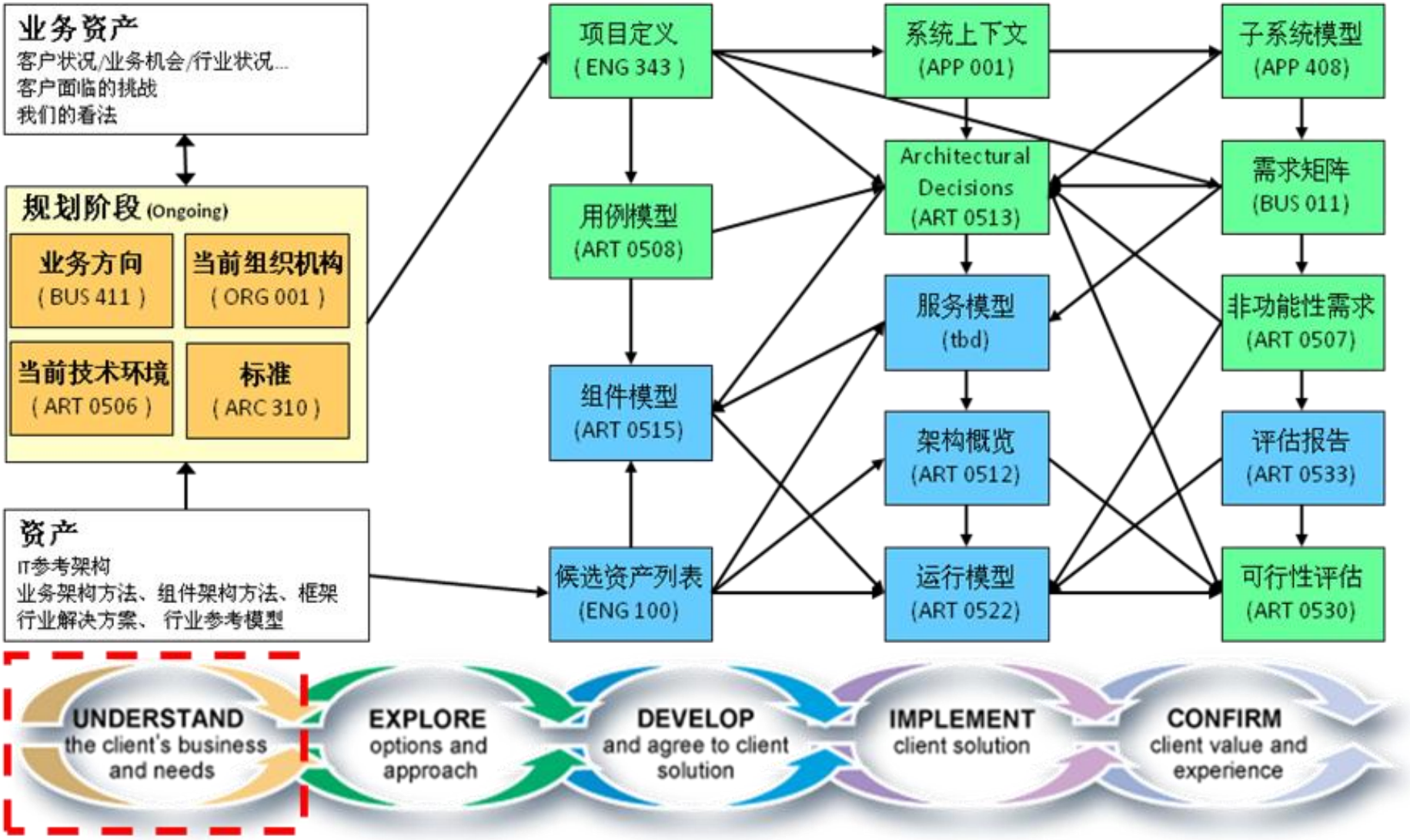
Plan	Pre-Sale Solution Design		Support Implementation and Confirm Value	
UNDERSTAND the client's business and needs	EXPLORE options and approach	DEVELOP and agree to client solution	IMPLEMENT client solution	CONFIRM client value and experience

TeAMethod Activities

KEY DECISIONS				
What value does the client want?	What options do we explore? Requirements?	What solution solves the problem?	How do we implement it successfully?	How will we enhance value?

OUTCOMES				
VALIDATED	QUALIFIED	WON	COMPLETED	New Opportunities Identified

不同阶段的工作产品


业务资产
客户状况/业务机会/行业状况...
客户面临的挑战
我们的看法
规划阶段 (Ongoing)
业务方向
(BUS 411)
当前组织机构
(ORG 001)
当前技术环境
(ART 0506)
标准
(ARC 310)
资产
IT参考架构
业务架构方法、组件架构方法、框架
行业解决方案、 行业参考模型
项目定义
(ENG 343)
用例模型
(ART 0508)
组件模型
(ART 0515)
候选资产列表
(ENG 100)
系统上下文
(APP 001)
Architectural Decisions
(ART 0513)
服务模型
(tbd)
架构概览
(ART 0512)
运行模型
(ART 0522)
子系统模型
(APP 408)
需求矩阵
(BUS 011)
非功能性需求
(ART 0507)
评估报告
(ART 0533)
可行性评估
(ART 0530)
UNDERSTAND
the client's business and needs
EXPLORE
options and approach
DEVELOP
and agree to client solution
IMPLEMENT
client solution
CONFIRM
client value and experience

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

系统学习带来的好处 - 我们的方法论课程

群体经验和智慧>个人

来报名吧·你不会失望

学习-成长-共享

定位与选择

架构师的职级构建
架构师分类与认证流程

架构原理
方法论
行业知识

- 前瞻性
- 知识管理
- 拥抱开源

有了智慧的人，才有智慧的产品，…智慧的地球

- 技术是核心，只有好的产品才能成功，不要迷信商业模式和营销
- 跟踪新的学术研究成果
- 拥抱开源，平台与云计算
- 参加专业、高质量的会议开拓眼界、广交同道、扩圈子
- 社会化媒体、专业社区
- 向互联网企业学习，别被淘汰，移动互联的未来
- 心胸开阔，善于沟通，不偏激
- 时间管理
- 知识管理
- 项目管理
- 行业知识的积累
- **定位与选择** - 做生态链上有价值的一环-方向比努力重要

Agenda

参考了以下资料完成(In Print, On The Web and Conference)

1. ***Grady Booch***: "*How Does It Work? The Architecture of Watson*", *Innovate2011.*
2. **"**Building Watson" in **AI Magazine**, vol. 31 (3).
3. **"**Unstructured Information Management in the *IBM Journal of Research and Development, vol. 43 (3), 2004.*
4. ***Final Jeopardy****: Man vs Machine and the Quest to Know Everything* by Baker.
5. **IBM** @ http://www.ibm.com/innovation/us/watson/
6. **N**ova @ http://www.pbs.org/wgbh/nova/tech/smartest-machine-on-earth.html
7. **TED** @ http://www.ted.com/webcast/archive/event/ibmwatson
8. **Wiki/Watson** @ http://en.wikipedia.org/wiki/Watson_(computer)

IBM Watson

Mission

- "We are using information as it exists and making the computer smarter in analyzing that content to compute answers."

– Dr. David Ferrucci, Principal Investigator for the Watson project

系统描述

- Watson是IBM制造的电脑问答（Q&A）人工智能系统，在比赛中用以挑战对人类自然语言的处理能力。

- 是一个集高级自然语言处理、信息检索、知识表示、自动推理、机器学习等开放式问答技术的应用系统，并且基于为假设认知和大规模的证据搜集、分析、评价而开发的DeepQA技术。

- 沃森是一台专为复杂分析而优化设计的系统，整合大规模并行处理器POWER7和IBM DeepQA 软件使其能在3秒内回答危险边缘的问题成为可能。

统计数据

开发团队 25 人

项目期限 4 年

软件 1,000,000+ SLOC

700K Java, 300K C++, plus other bits

~ 130 components

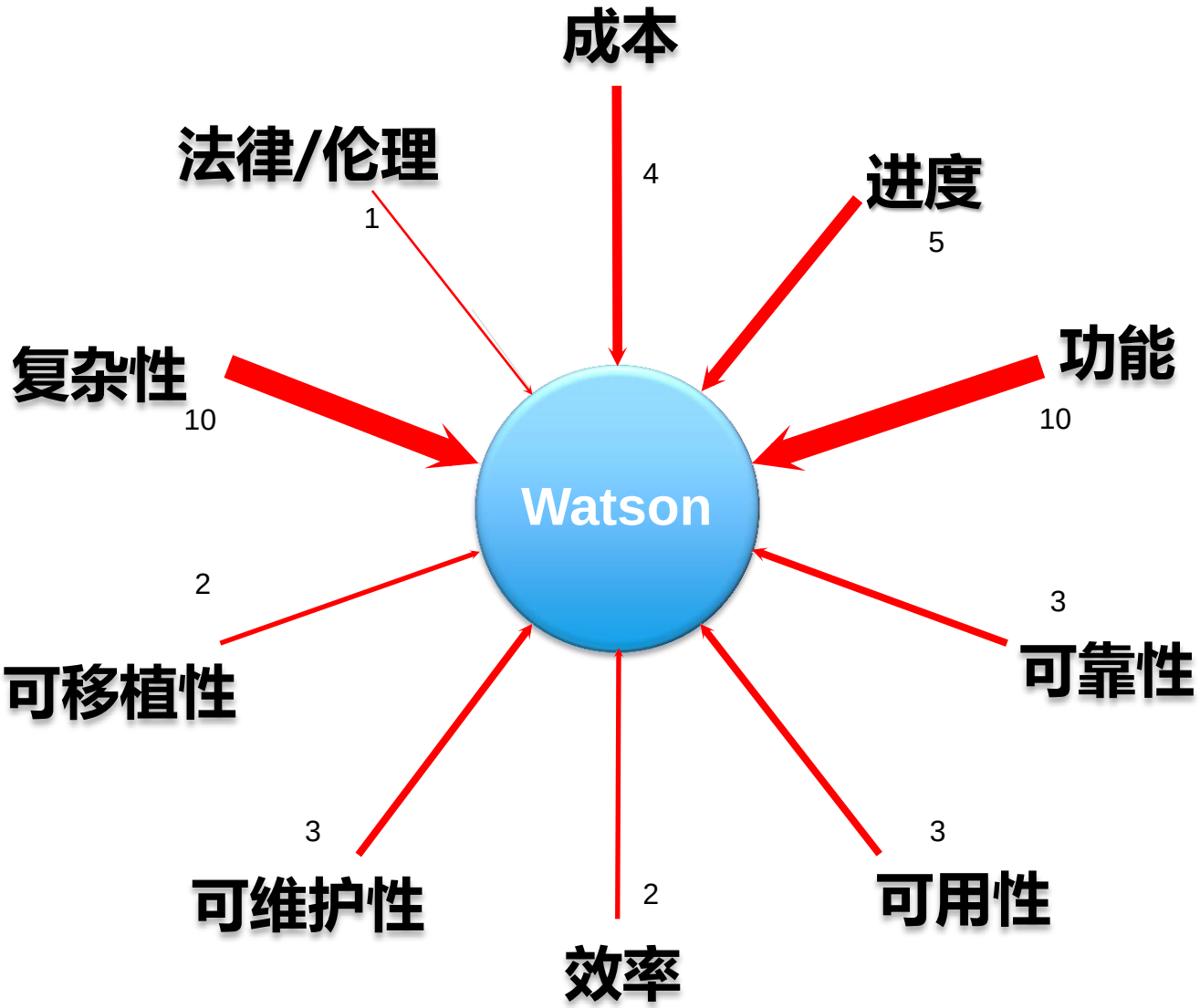
硬件 10组 90台 IBM Power 750™ servers

2880 Power7 cores @ 80+ TFLOPS

16 TB memory

10 Gbps network

约束



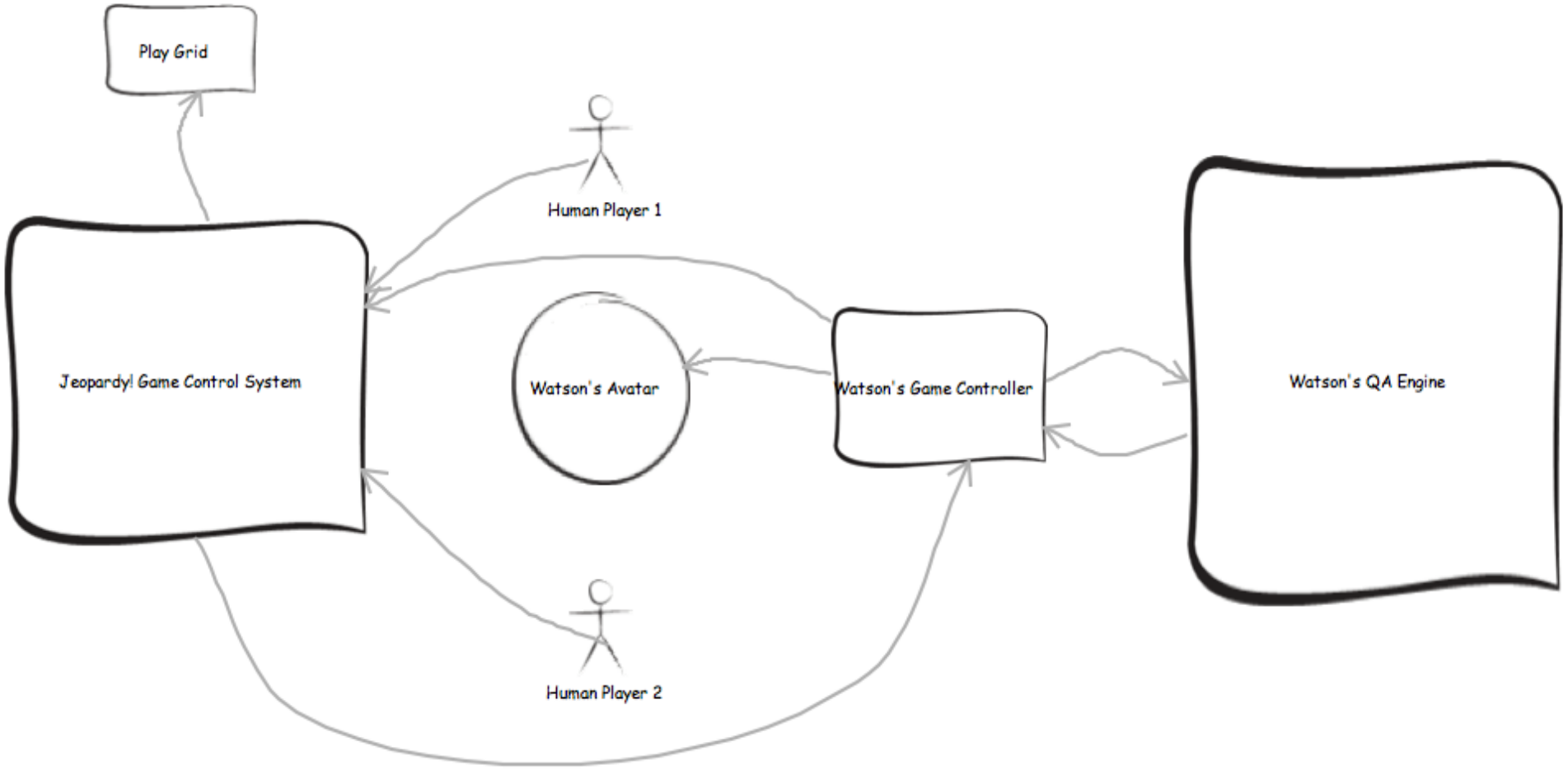
规则

- 危险边缘游戏中，所有选手（也包括沃森），必须等到主持人将每个线索念完，然后就绪灯亮起，第一个按下抢答器按钮的人可以获得回答问题的机会。

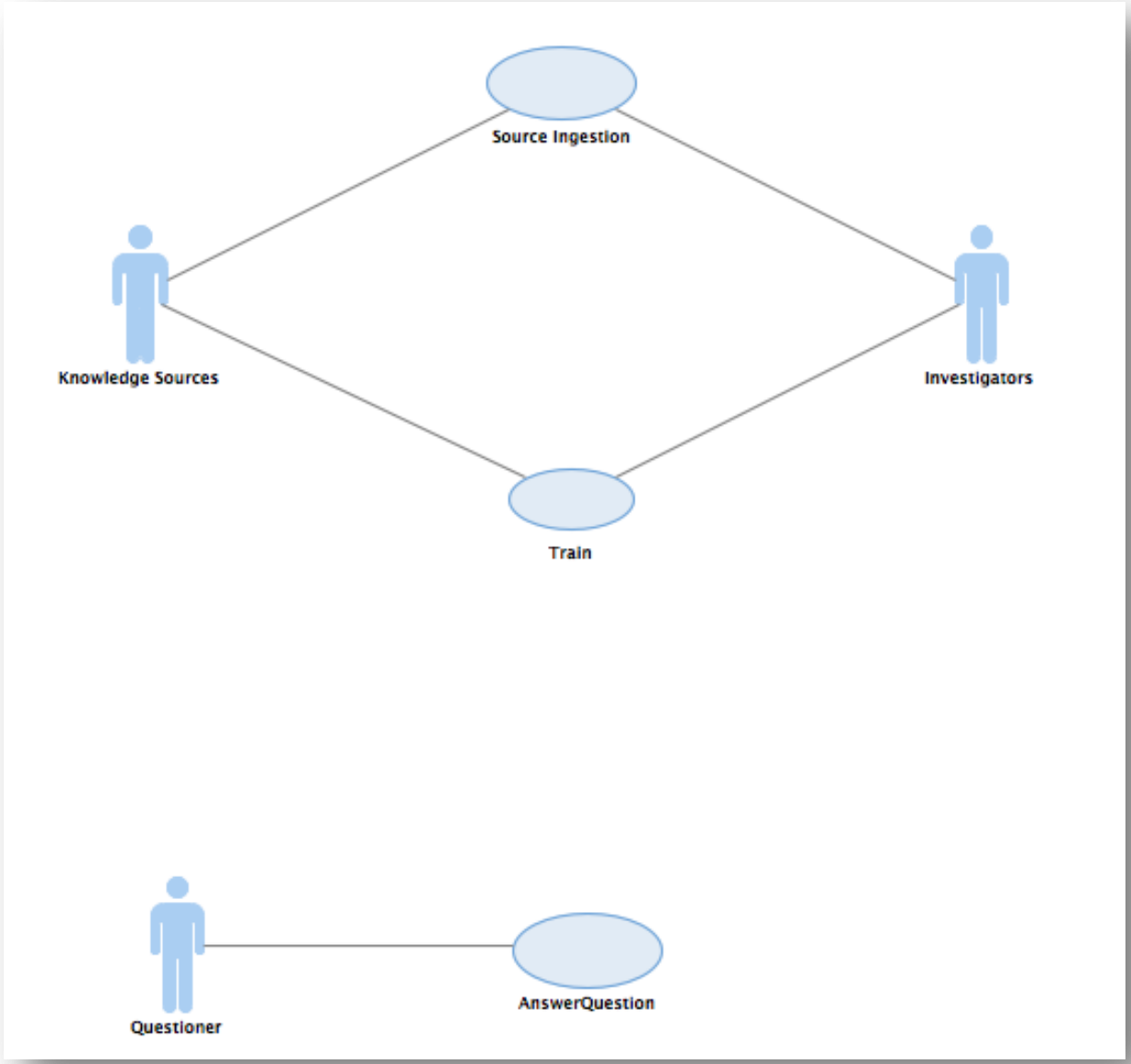
基本工作原理

- Watson像接收电子文本一样得到这些线索的同时这些线索也显示给人类选手。然后它会将这些线索解析为不同的关键字和句子片段，这样做是为了查找统计相关词组。
- Watson最革新的并不是在于全新的操作算法，而是能够快速同时运行上千的自然语言分析算法来寻找正确的答案。算法找出的相同答案越多，沃森就越肯定答案正确。一旦沃森发现一个潜在的解决方法，并且这个解决方法有效，它就会核对数据库来确定答案。

架构: System Context



用例Use Cases


Source Ingestion
Knowledge Sources
Investigators
Train
Questioner
AnswerQuestion

关键设计决策： 技术决策(Technical)

- 使用pipe and filter 架构风格。
- 获取和使用海量的异构数据源。
- 考虑多种可能的侯选答案。
- 检索与评估支持每个侯选答案的多种证据。
- 从多个维度评估证据的置信度。
- 结合使用机器学习的证据。
- 构建于UIMA(非结构化信息管理架构)。

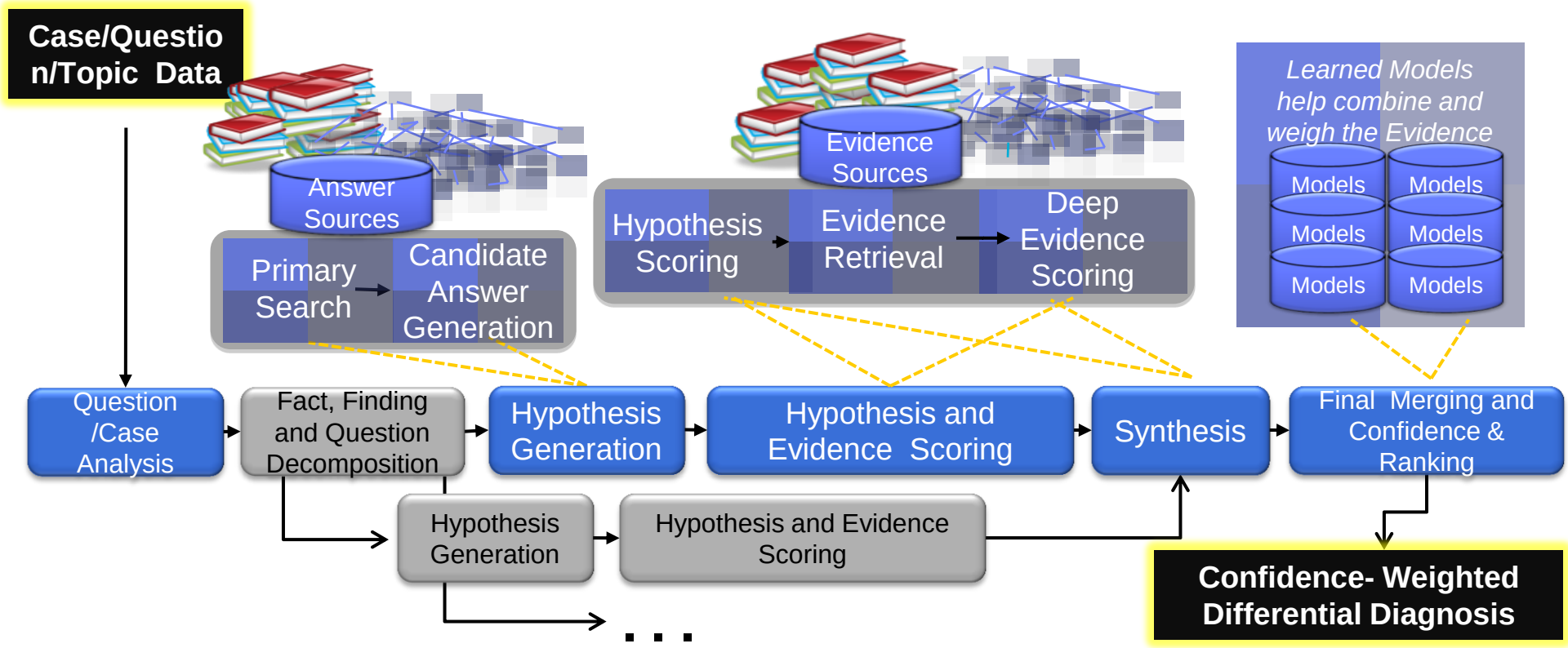
关键设计决策： 运行视角(Operational)

- 根据特定的部署场景，允许数据位置可配置。
- 从大规模逻辑并行映射到大规模部署并行 (易于重构的机制)。

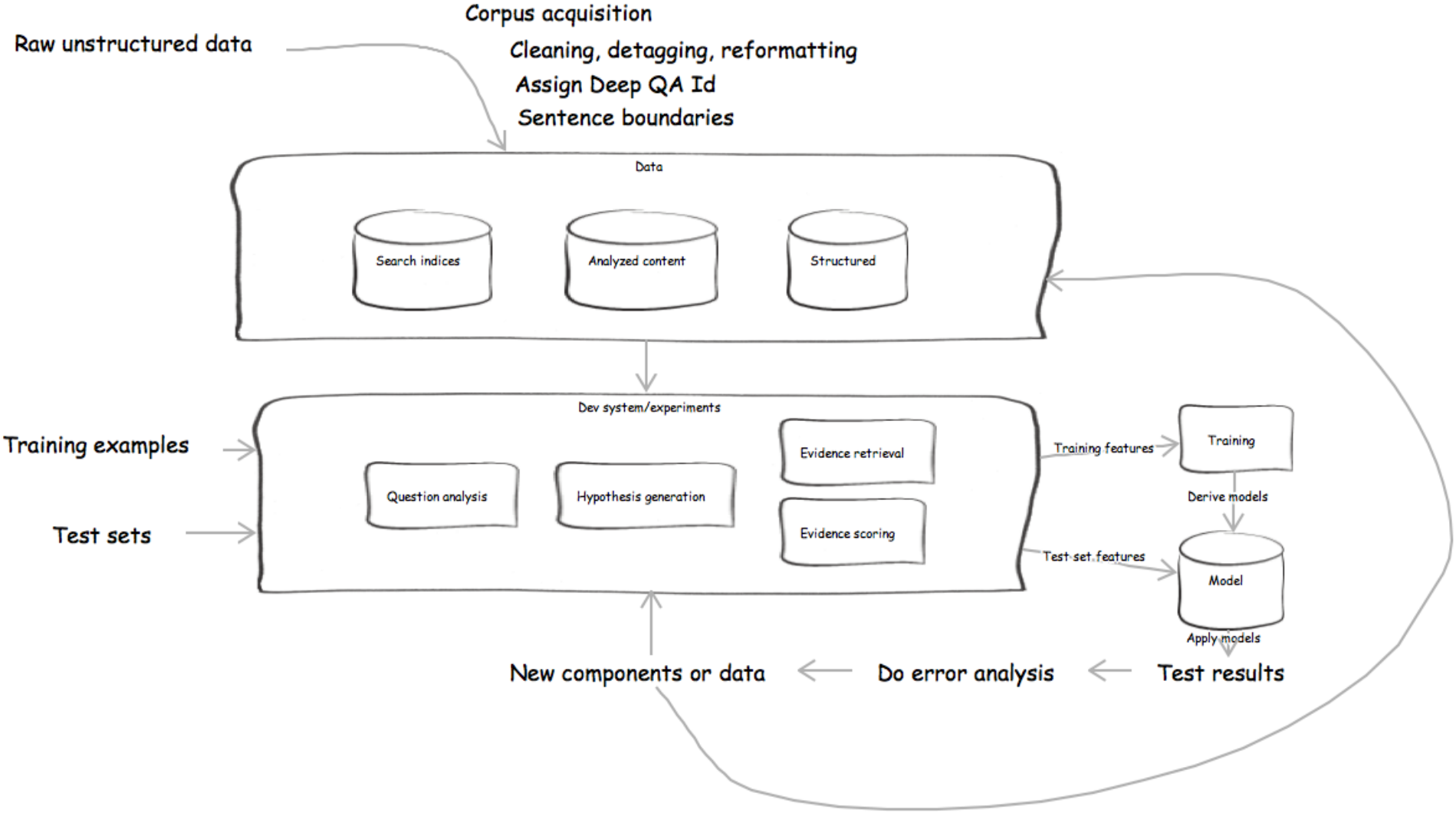
关键设计决策： 方法视角(Methodological)

- 构建鲁棒的端到端的衡量标准。
- 维护数据、算法和流程的重要元数据信息。
- 构建工具分析Watson的运行和思考过程。

关键组件DeepQA的高层架构



DeepQA架构：知识获取 / 回答



原始的非结构化数据→语言资料库采集(清洗，重新格式化，分配DeepQA的ID，句子边界→搜索索引、分析内容、结构化→问题分析、假设创建、检索循证、证据得分

DeepQA组件的核心功能1：知识获取

- **沃森的信息来源包括各种授权的百科全书、字典、词典、新闻和文学作品。沃森也使用如下的数据库、分类学和本体论。**
- **Wikipedia/Wikiquote/Wiktionary/Wikibooks** (The Free Encyclopedia) @ http://wikipedia.org
- **YAGO2** (A Spatially and Temporally Enhanced Knowledge Base from Wikipedia) @ http://www.mpi-inf.mpg.de/yago-naga
- **Dbpedia** (Extracting Structured Information from Wikipedia) @ http://dbpedia.org
- **WordNet** (A Lexical Database for English) @ http://wordnet.princeton.edu
- **Web expansion of many primary sources.**
- 这些知识一般可以总结为：【主本体】 <关系> 【辅助本体】 这样的三元组。DeepQA爬取和整理很多这类的三元组用来直接获取候选答案，替换子句的问题，确定答案类别等方面。

测试和训练样本数据

- **J! Archive** (A Fan-created Archive of Jeopardy!)

  @ http://www.j-archive.com

Jeopardy节目20年来的所有题目

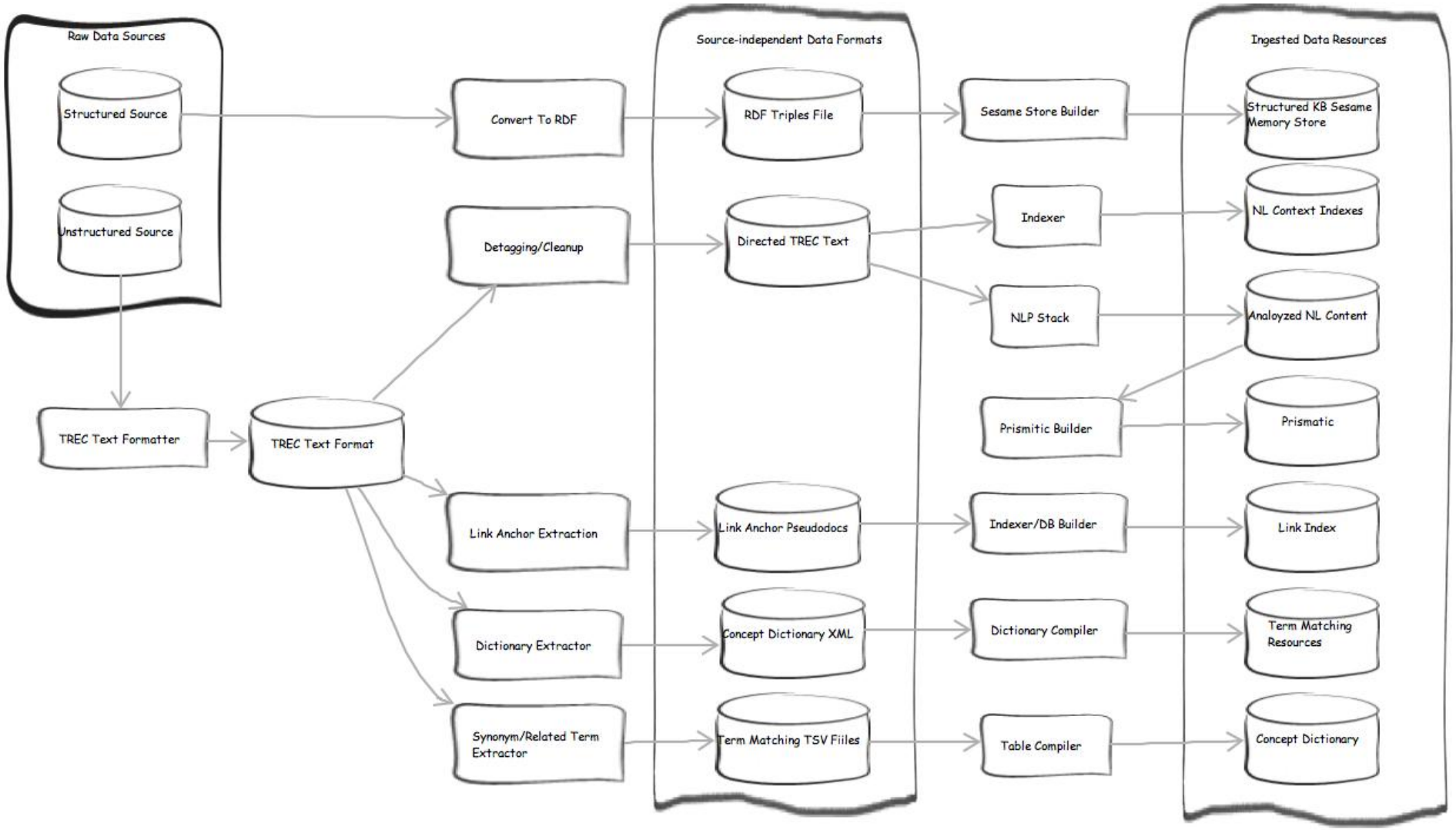
DeepQA的核心功能2：自动问答

1. 对用户的问题进行词法分析，句法分析，问题分类，实体识别，各种语法语义标注，角色标注，答案类型分析等
2. 判断问题的多个子问题之间的关系，是否分解成多个问题并行处理或应用其他处理策略
3. 在对答案库进行搜索后，抽取出可能的答案，替换到问句中组成陈述句类型的假设集（把答案填充到问题的疑问部分，改为陈述句）
4. 过滤掉一部分可能性不高的假设
5. 以样例库为标准，综合检验各个假设的置信度有多高，并给出置信度值
6. 综合各种渠道的信息，总的给各个答案一个排序。

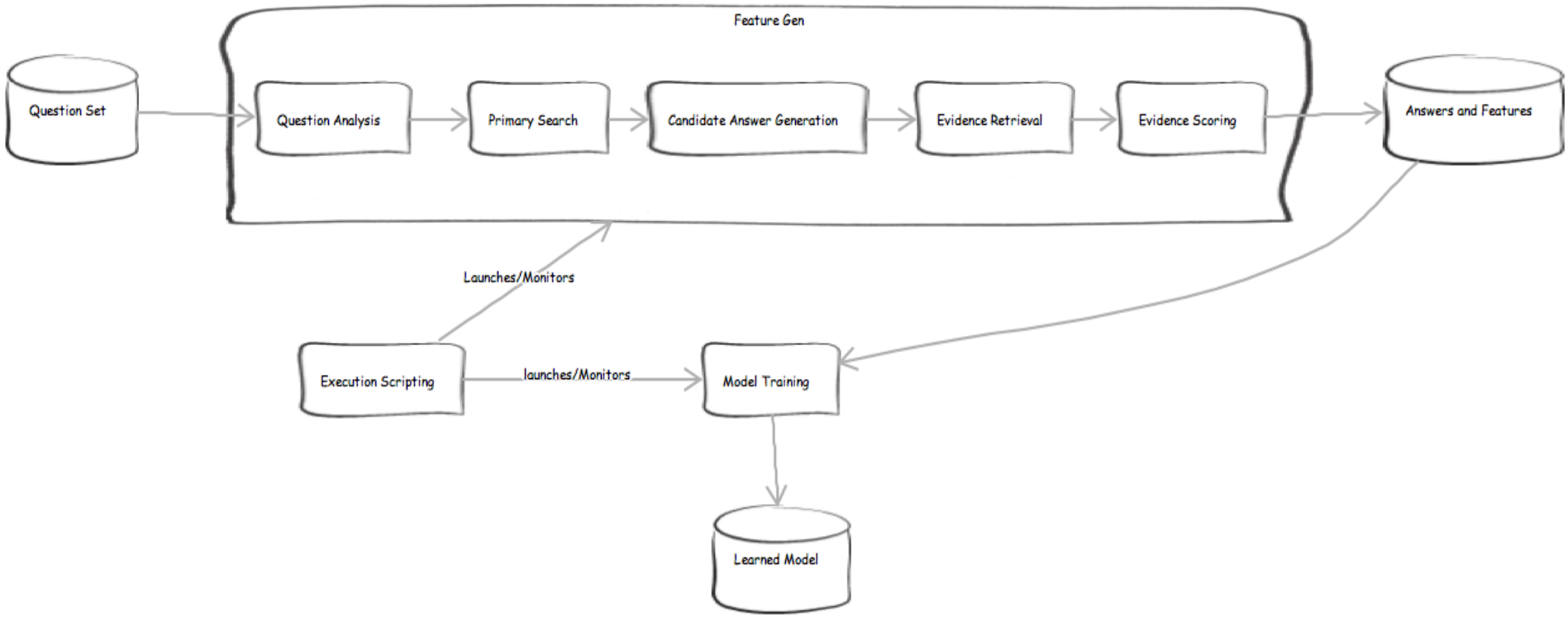
架构：样本获取

- 分析例子问题和特征域.
- 资源收集, 包括数据库, 分类标准, 本体.
- 扩展语料库.
  - 识别种子文档并检索相关文档.
  - 抽取自包含的有价值文本.
  - 关联度分析.
  - 合并置信度最高的信息到语料库.
- 推导运行时的数据集 (search indexes, hashmaps, triple stores, etc).
- 通过**Hadoop**运算使用UIMA annotations做NLP预分析.

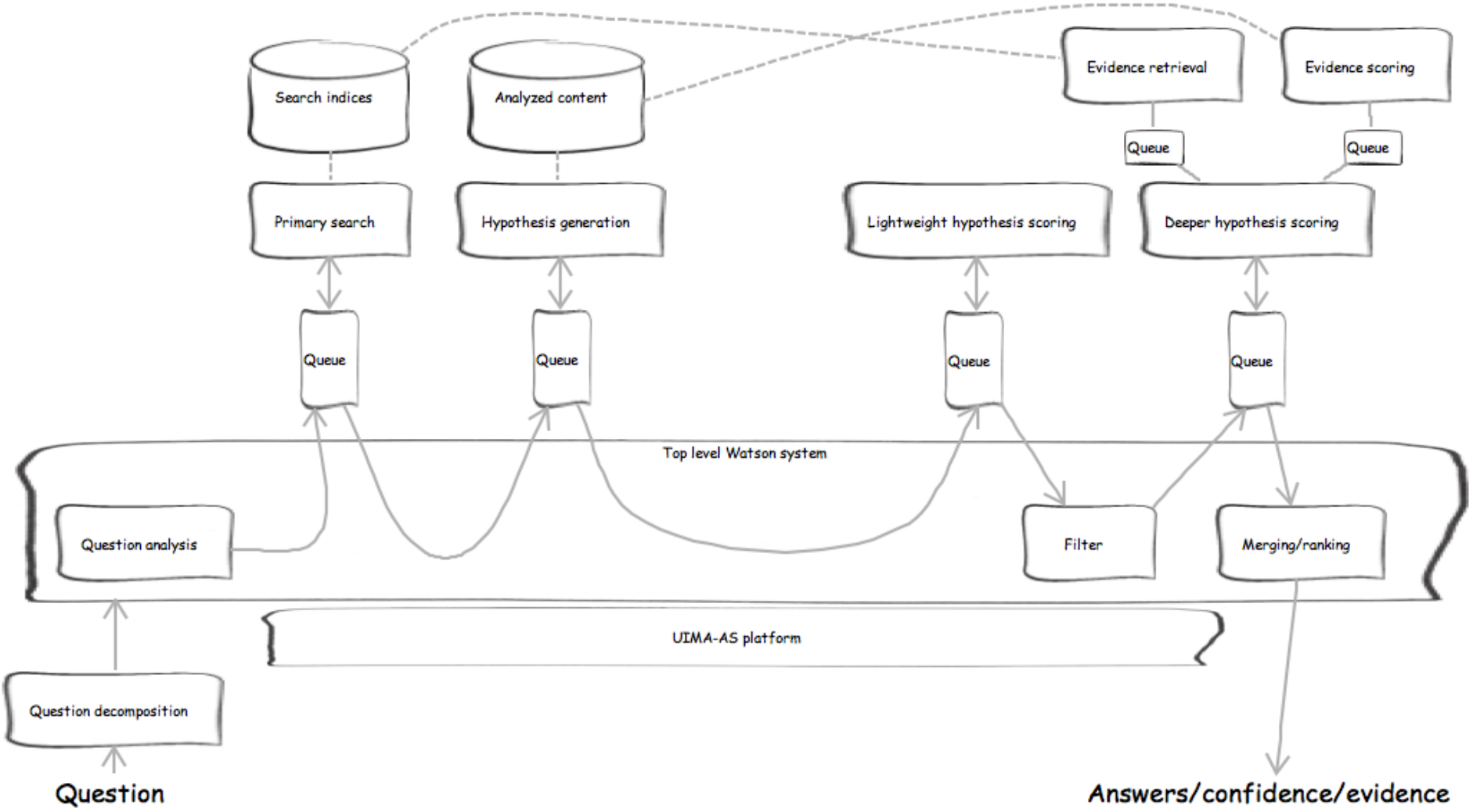
架构：样本获取


Raw Data Sources
Structured Source
Unstructured Source
TREC Text Formatter
TREC Text Format
Convert To RDF
Detagging/Cleanup
Link Anchor Extraction
Dictionary Extractor
Synonym/Related Term Extractor
Source-independent Data Formats
RDF Triples File
Directed TREC Text
Link Anchor Pseudodocs
Concept Dictionary XML
Term Matching TSV Fiiles
Sesame Store Builder
Indexer
NLP Stack
Prismitic Builder
Indexer/DB Builder
Dictionary Compiler
Table Compiler
Ingested Data Resources
Structured KB Sesame Memory Store
NL Context Indexes
Analoyzed NL Content
Prismatic
Link Index
Term Matching Resources
Concept Dictionary

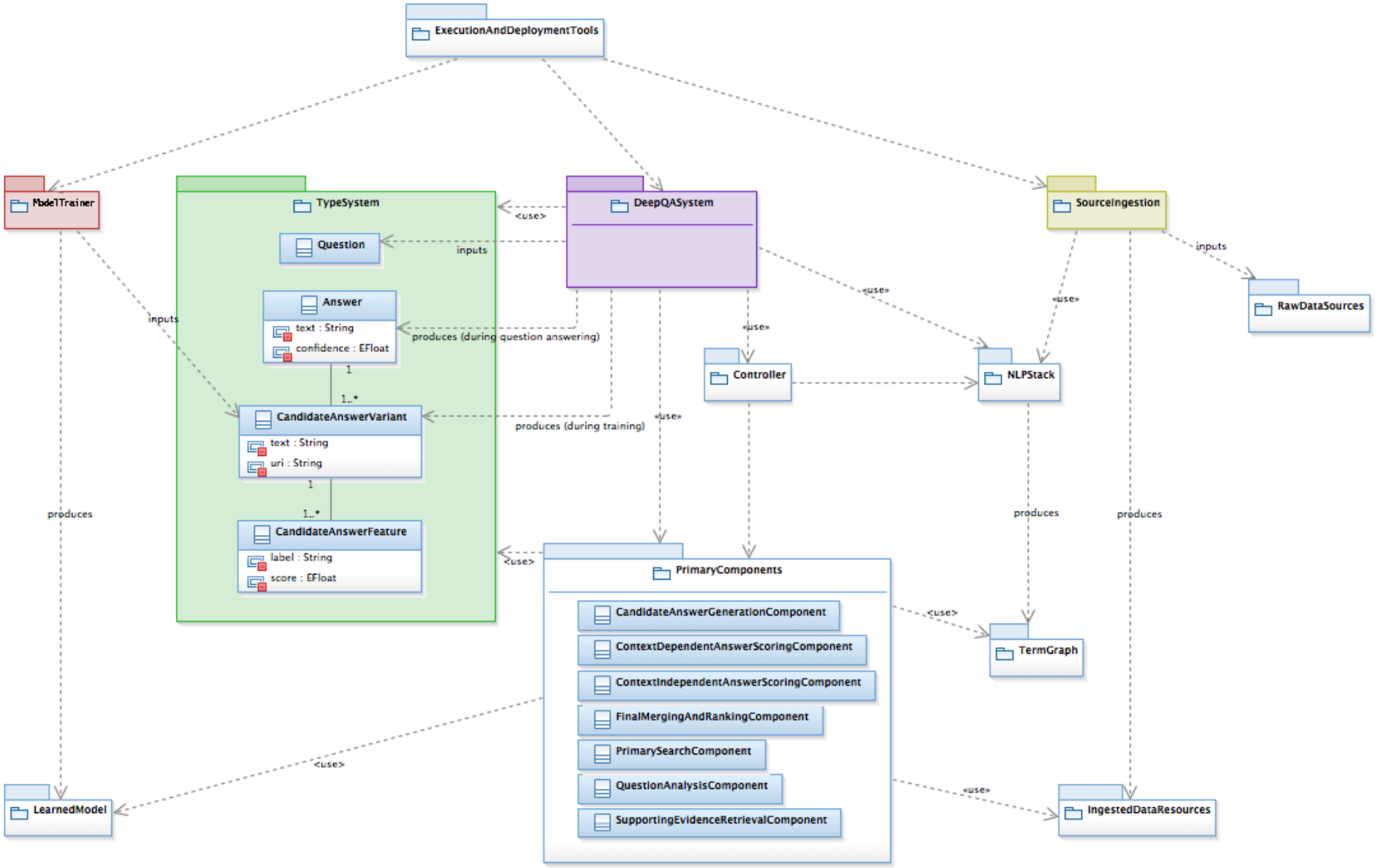
架构：Learn


Feature Gen
Question Set
Question Analysis
Primary Search
Candidate Answer Generation
Evidence Retrieval
Evidence Scoring
Answers and Features
Launches/Monitors
Execution Scripting
launches/Monitors
Model Training
Learned Model

架构：Play


Search indices
Analyzed content
Evidence retrieval
Evidence scoring
Queue
Queue
Primary search
Hypothesis generation
Lightweight hypothesis scoring
Deeper hypothesis scoring
Queue
Queue
Queue
Queue
Top level Watson system
Question analysis
Filter
Merging/ranking
UIMA-AS platform
Question decomposition
Question
Answers/confidence/evidence

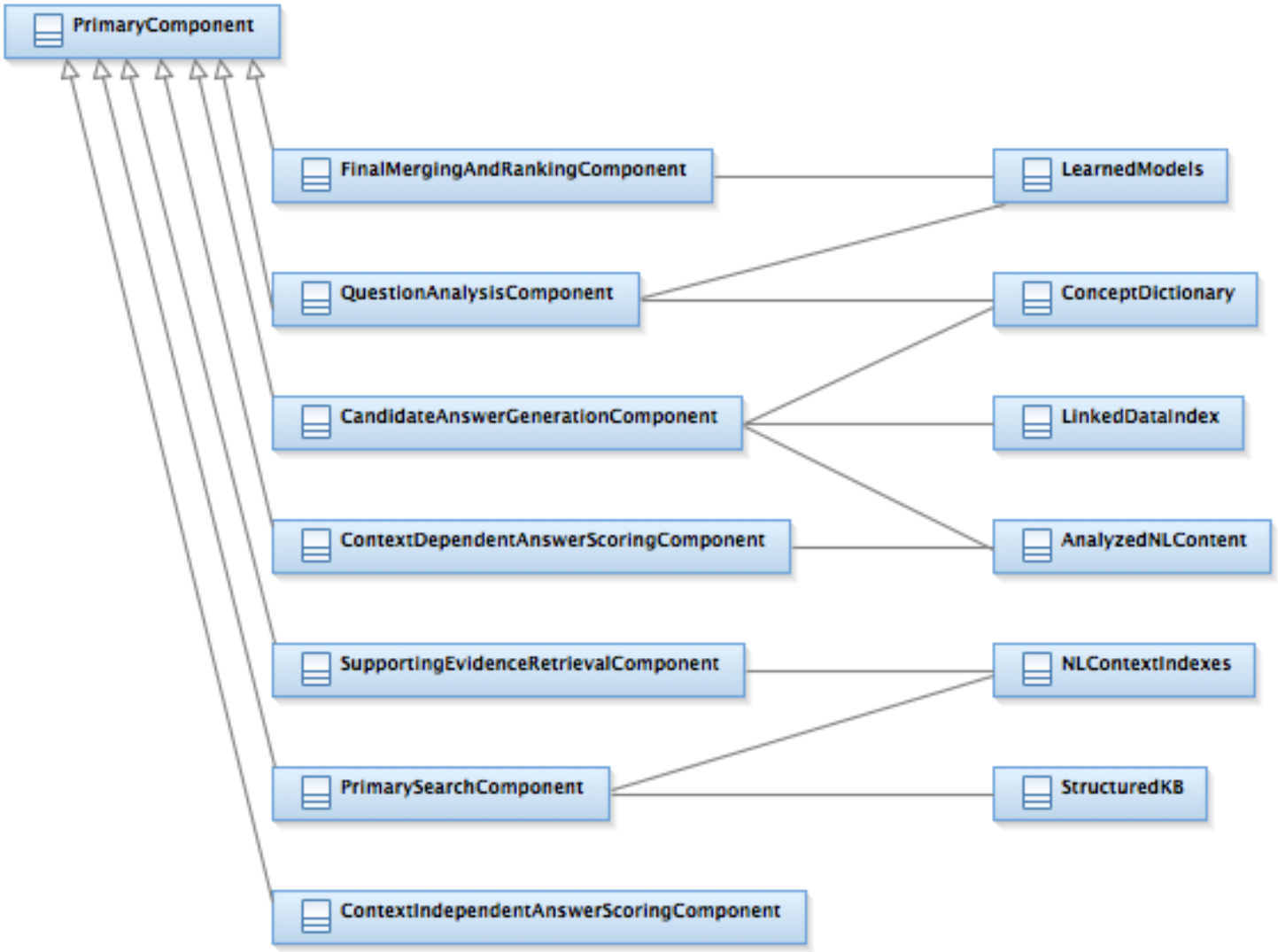
逻辑视图:核心子系统模型



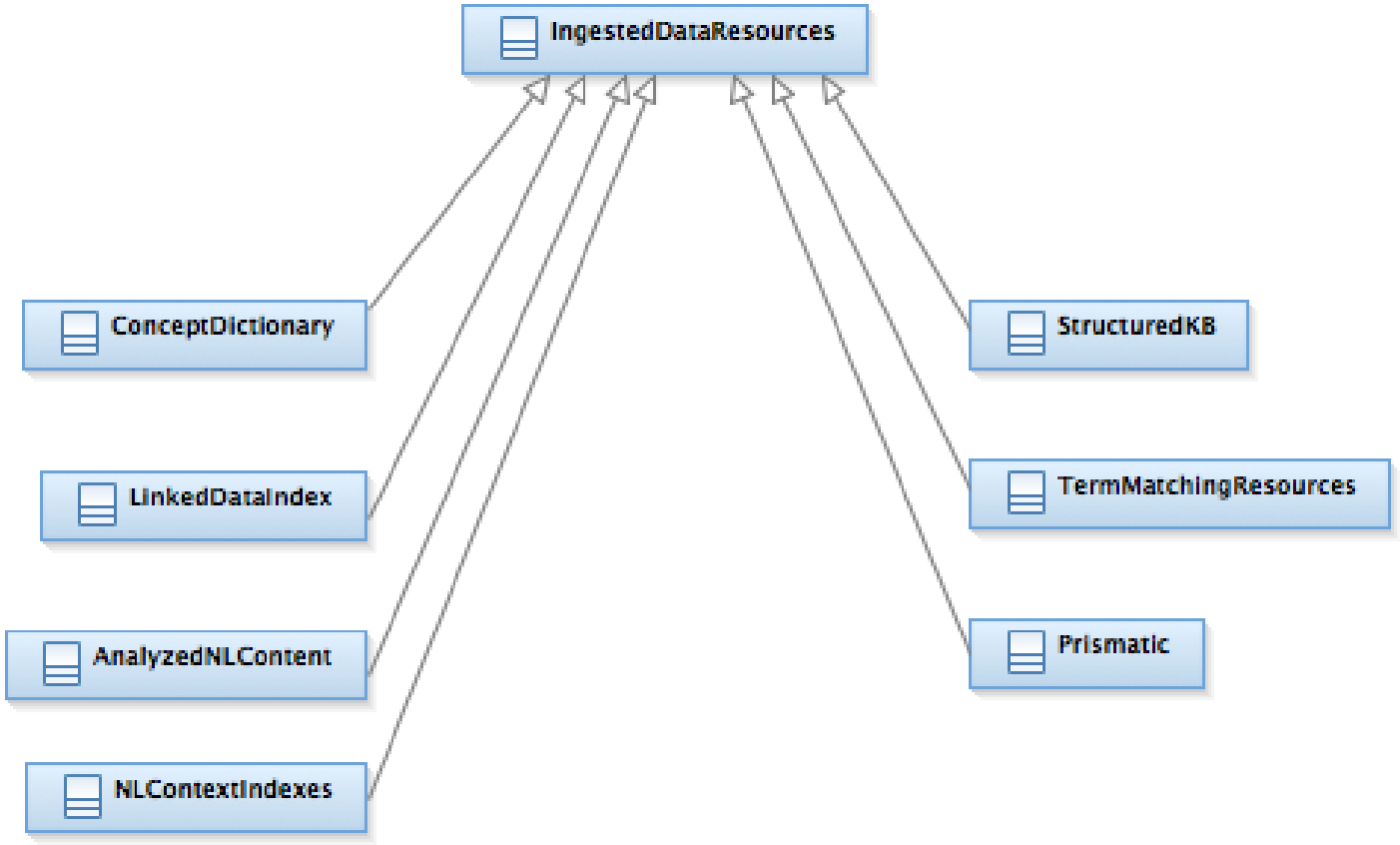
逻辑视图:关键特性

- UIMA Common Analysis Structure (CAS)
- CoreTypeSystem (aka Data Model).
- QuestionAnalysisTypeSystem.
- Terms.
- IngestedDataResources.
- PrimaryComponents.
- ModelTrainer.

逻辑视图： Primary Components



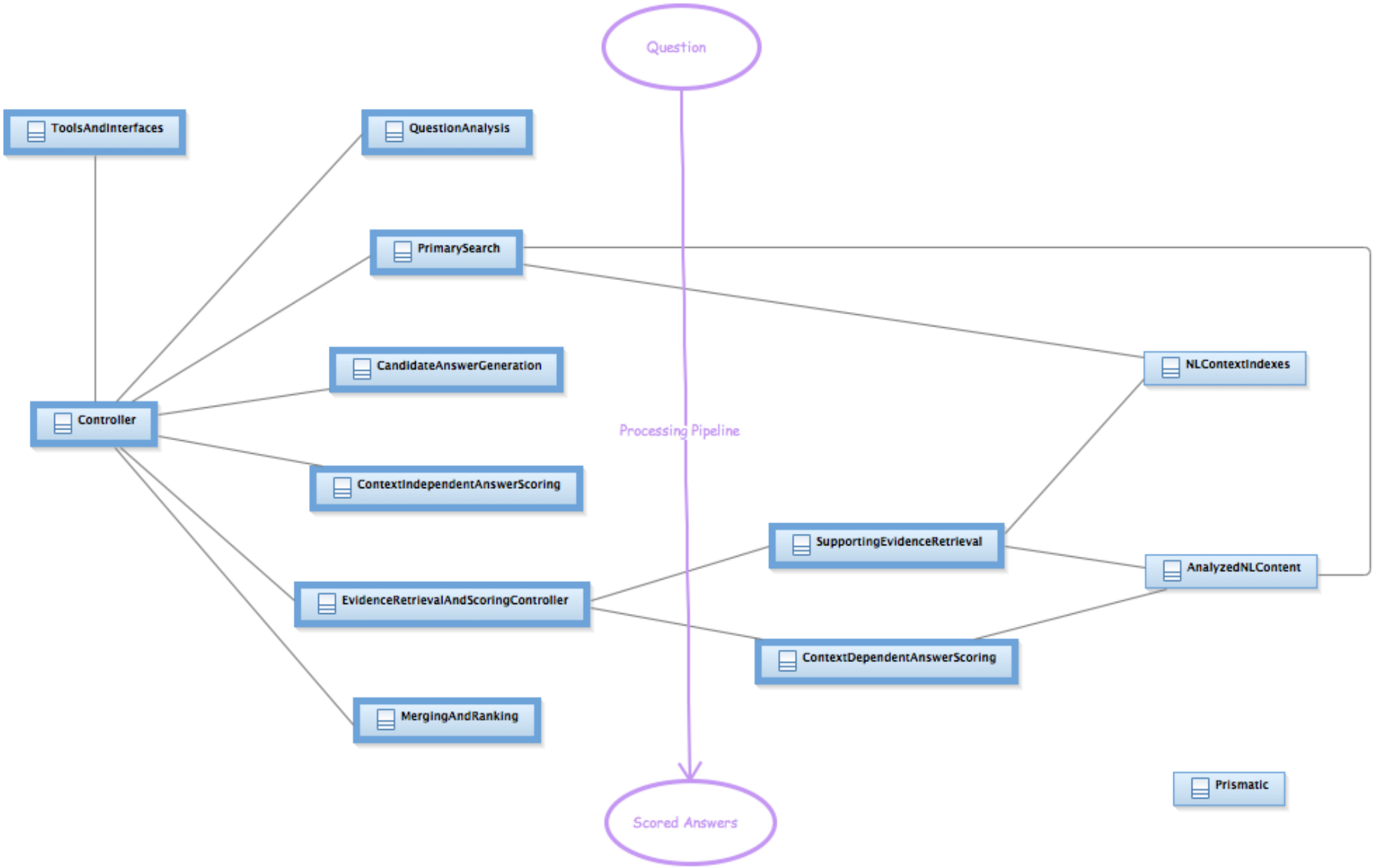
逻辑视图： 摄入的数据源


IngestedDataResources
ConceptDictionary
LinkedDataIndex
AnalyzedNLContent
NLContextIndexes
StructuredKB
TermMatchingResources
Prismatic

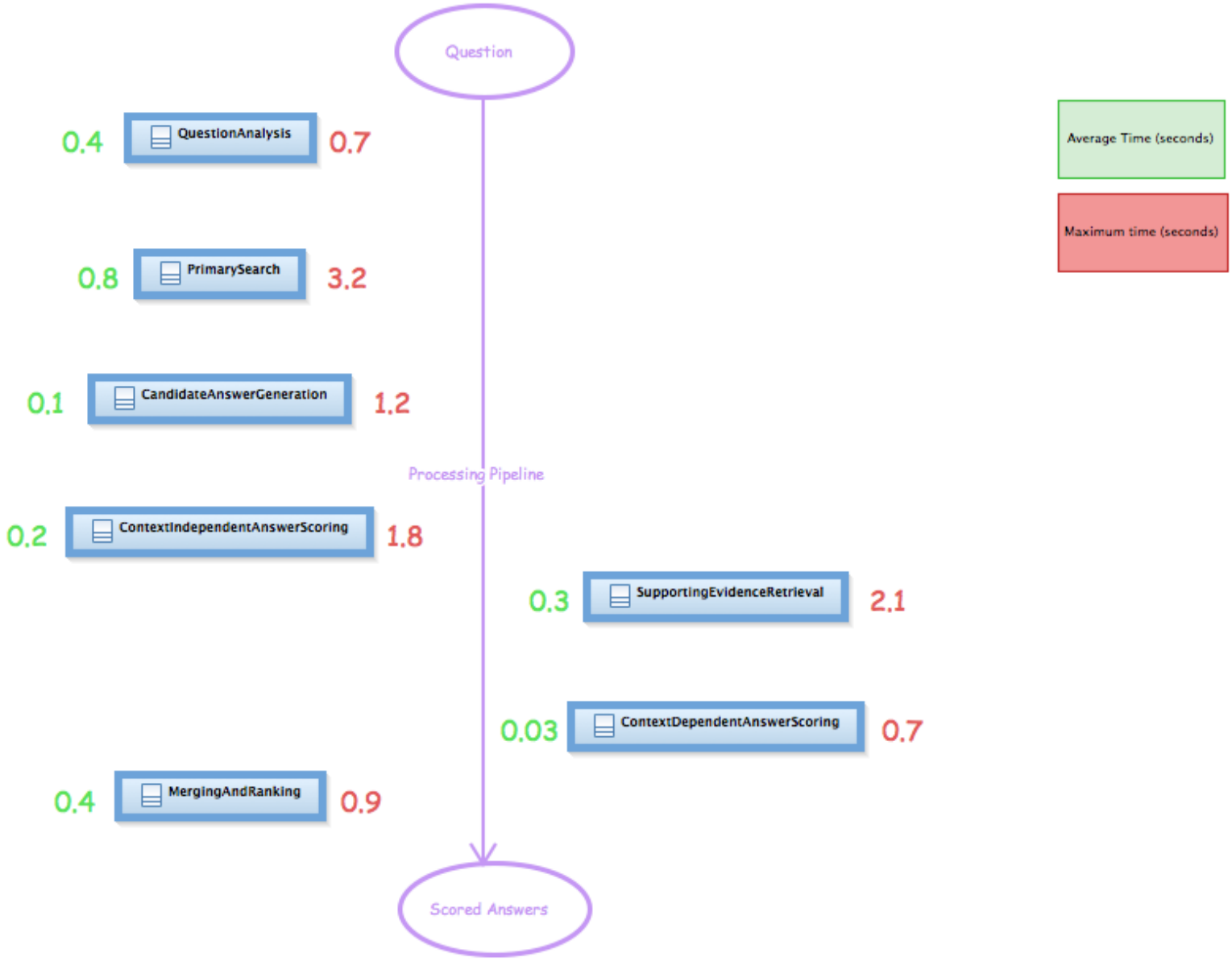
逻辑视图： 核心处理机制

- UIMA.
- Question analysis.
- Primary search - Indri 和 Lucene开源搜索引擎.
- Candidate answer generation.
- Shallow (content independent) and deep (context dependent) scoring.
- Merging and ranking.

Process View: Processing Pipeline



Process View: Timing



Process View: Key Mechanisms

- UIMA-AS.

  A set of general capabilities for achieving scale out, built upon UIMA.

- UIMA CAS Multiplier and CAS pools.

  Expand/consolidate CAS envelopes efficiently across multiple configurable flows.

- Three communication protocols:

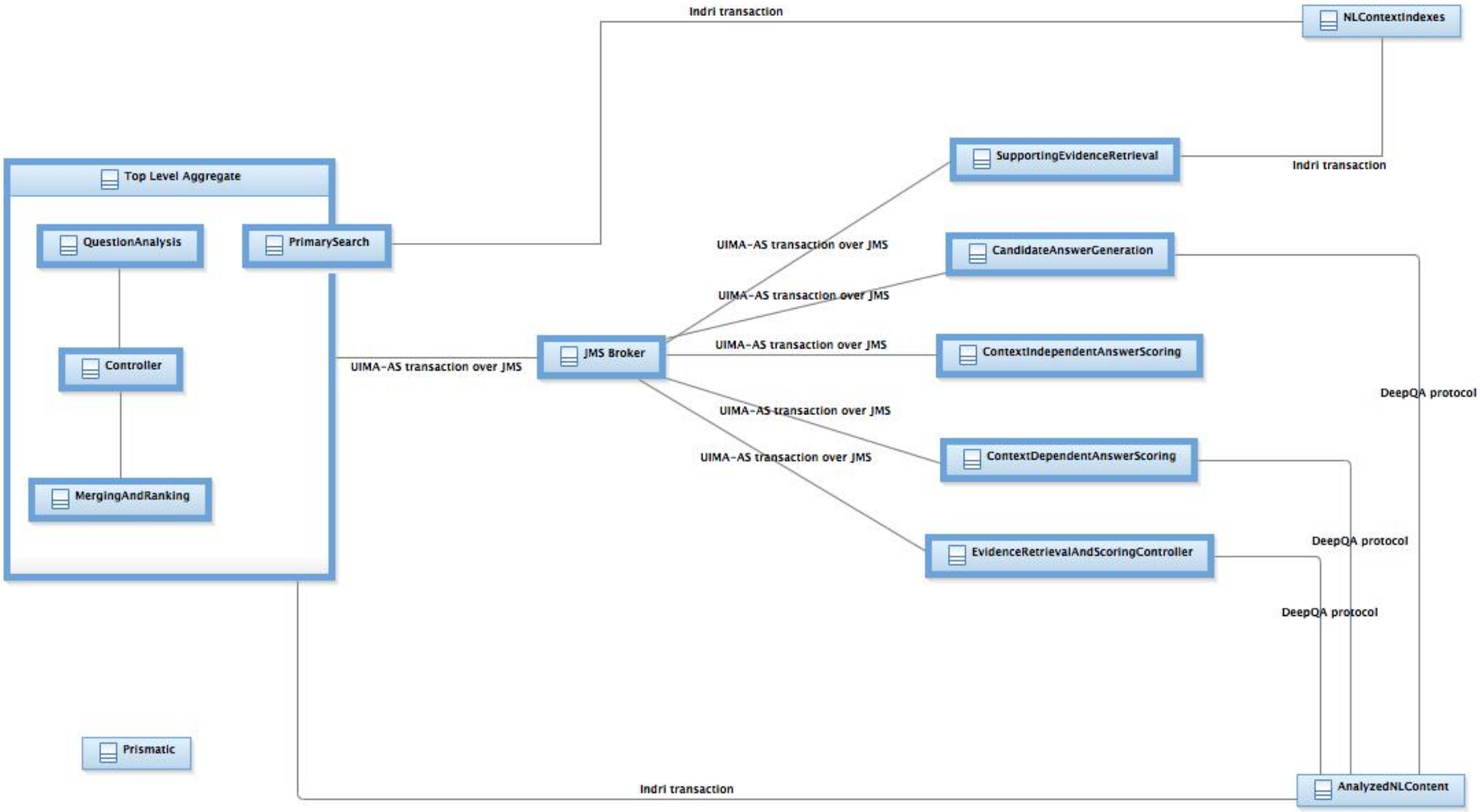
  UIMA-AS transactions across JMS.

  DeepQA protocol for accessing large in-memory datasets.

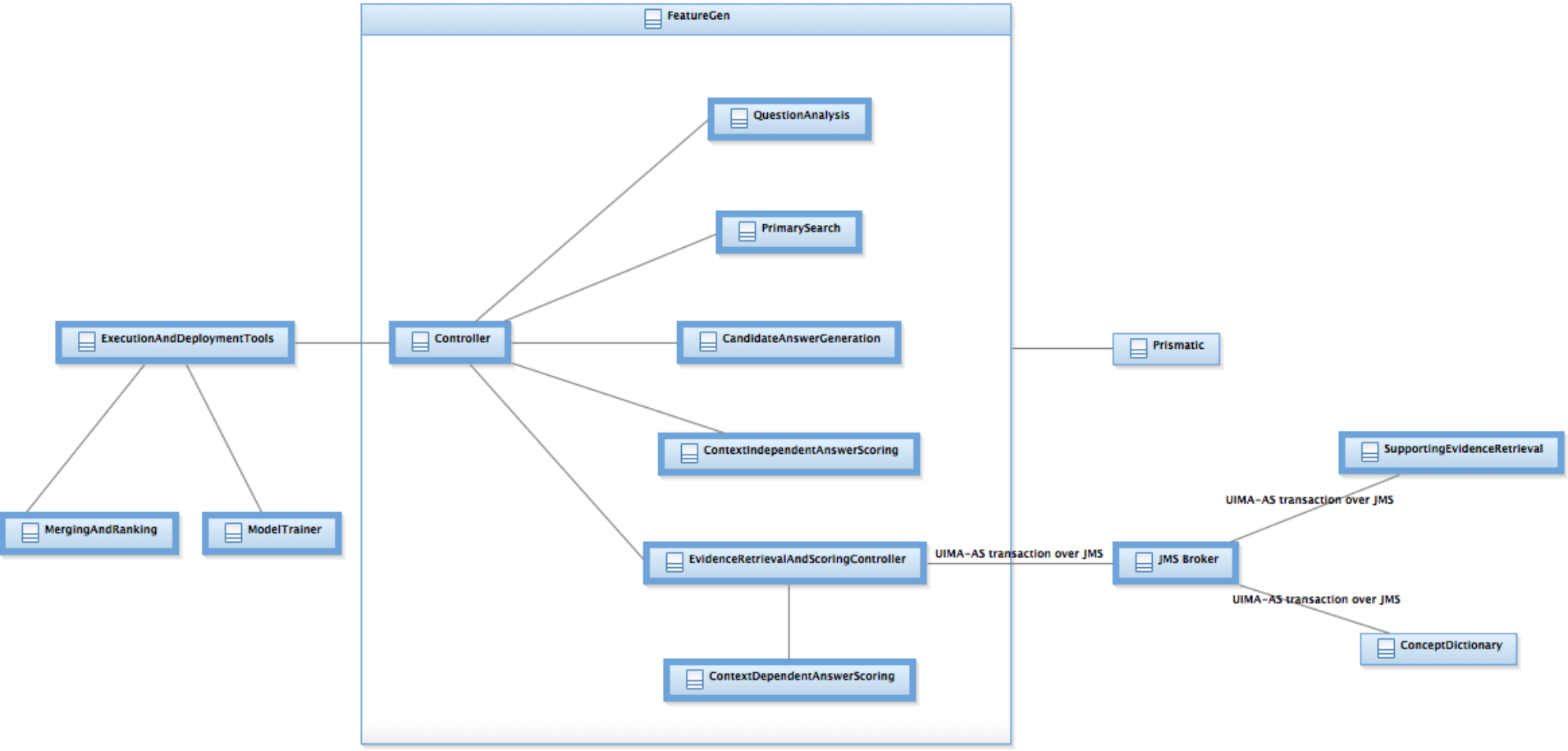
  Indri distributed search protocol.

http://uima.apache.org/doc-uimaas-what.html

Process View: Low-latency Production



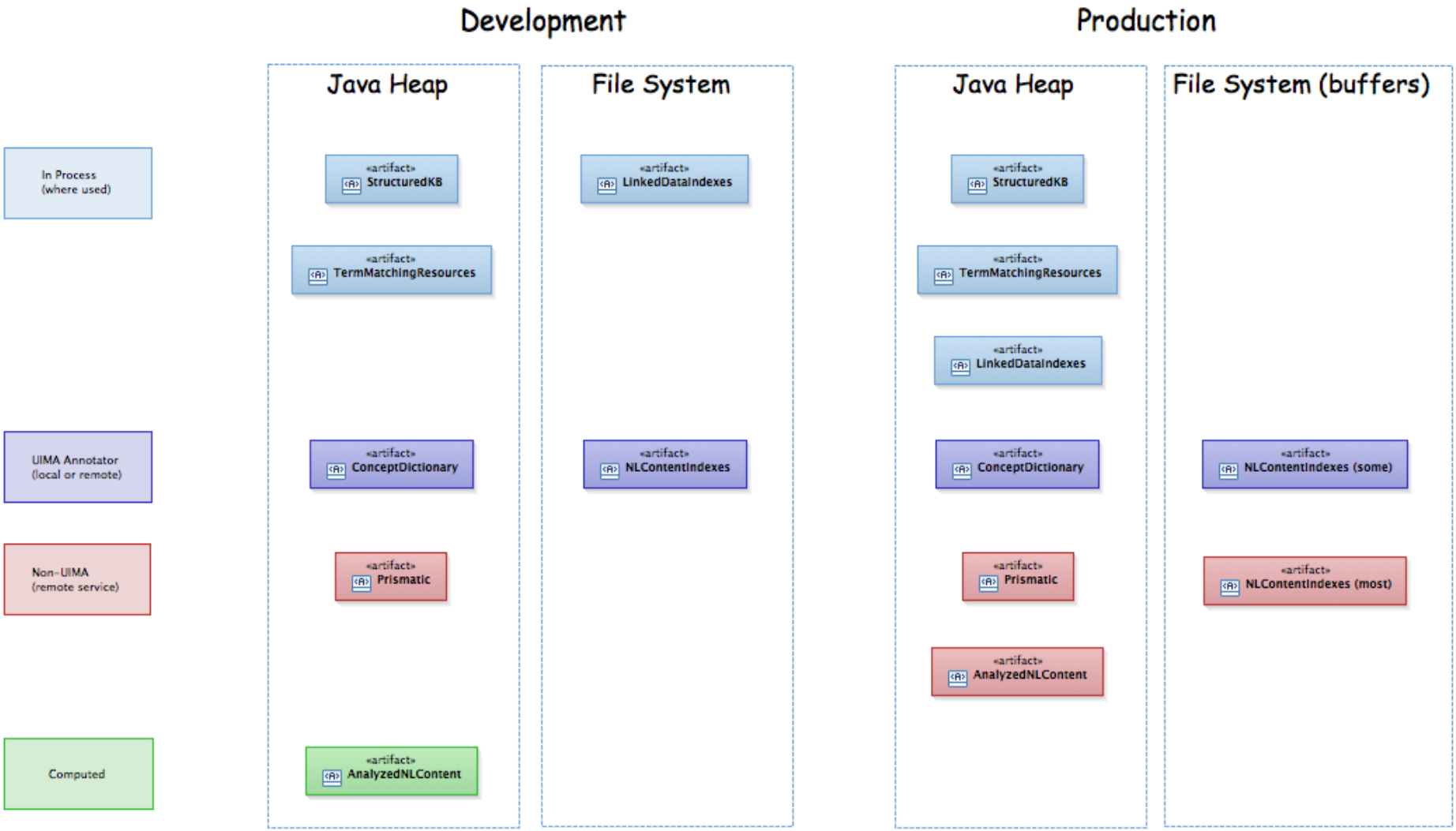
Process View: High-throughput Development



Component View: Code Layering

<table>
<tr><td colspan="5">Execution Scripts/Deployment Tools
(bluej.system.ant/bluej.system.as)</td></tr>
<tr><td colspan="4">DeepQASystem
(bluej.system)</td><td>Development Tools
(parts of bluej.tools)</td></tr>
<tr><td colspan="5">InternalComponents/TermSource/TermMatcher
(sai.matcher/bluej.container/
sai.disambiguation/etc)</td></tr>
<tr><td>Source Ingestion
(bluej.tools_corpus_processing/blue.corpus_processing)</td><td colspan="3">Evidence Interfaces/Resource Interfaces
(bluej.ksp/bluej.rdf/bluej.prismatic/
bluej.content_server/bluej.spatail)</td><td>Controller
(bluej.core)</td></tr>
<tr><td>DeepQA Utilities
(bluej.utill)</td><td>Type System
(bluej.model)</td><td>NLPStack</td><td>TermGraph
(sai.logical_form.kr)</td><td>Base Tools
(bluej.tools.corpus_processing/bluej.corpus_processing)</td></tr>
<tr><td colspan="3">General Utilities
(sai.utilities/3rd party JARS)</td><td colspan="2">UIMA
(uima)</td></tr>
</table>

Component View: Data Storage

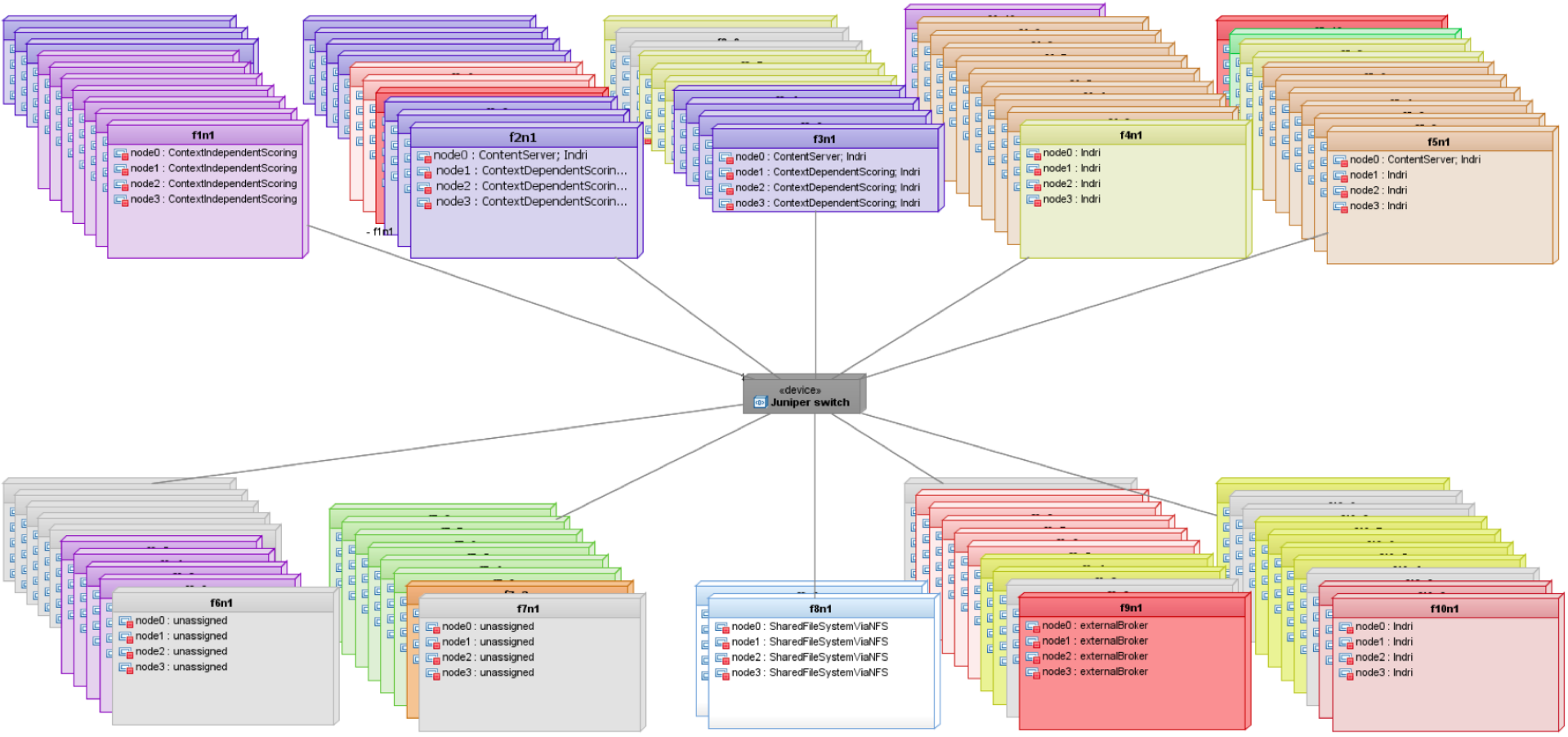


Deployment View: 硬件(Server、网络和存储)

- 90台 Power 750™ 服务器.
  - 4 Power7 processors/server.
  - 8 cores/processor.
- 2880 POWER7 内核
- POWER7 3.55 GHz 芯片
- 16 TB memory和4 TB clustered storage
- IBM SONAS 存储集群@ 20 TB.
- Juniper switch @ 10 Gbps.
- 2 20-air conditioning units(40吨重).

http://www.hypeframework.org/blog/content/ibm-watson-and-the-jeopardy-challenge/

Deployment View: Watson（10组机架）



CandidateGeneration	ContentServer*	ContextDependentScoring*	ContextIndependentScoring	Supporting Evidence Retrieval / EvidenceRetrievalAndScoringController	FileSystem
Indri (NIContentIndexes)	Lucene (PrimarySearch + NIContentIndexes)	Prismatic	TopLevelAggregate	Other	Unassigned

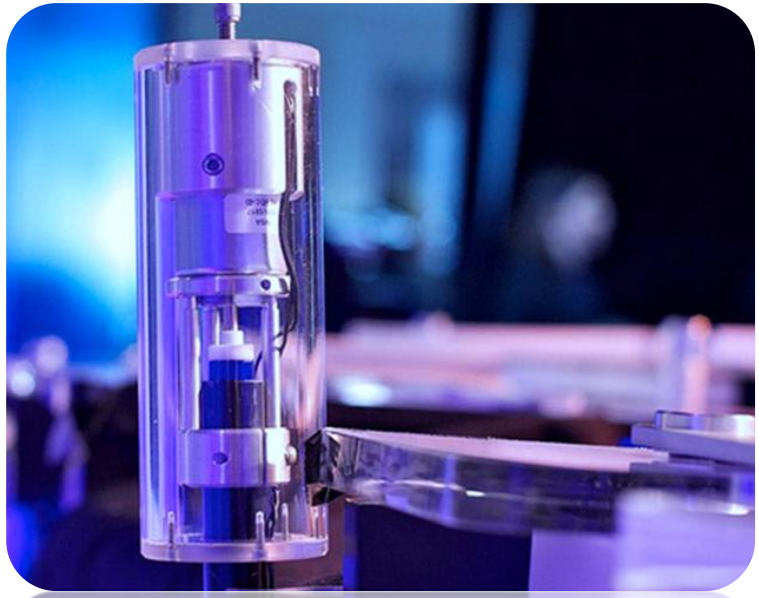
* denotes Indri processes also deployed on same node

Deployment View：软件环境

- 应用软件基于Java和C++编写
- 使用Apache Hadoop框架做分布式计算
- Apache UIMA框架
- IBM DeepQA软件
- IBM InforSphere BigInsights Platform(Hadoop商用实现)
- SUSE Linux Enterprise Server 11
- Indri 和 Lucene搜索引擎
- 100多种NLP算法实现
- Nuance的语音识别系统

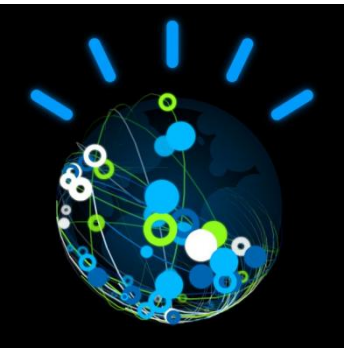
Deployment View: 促动装置

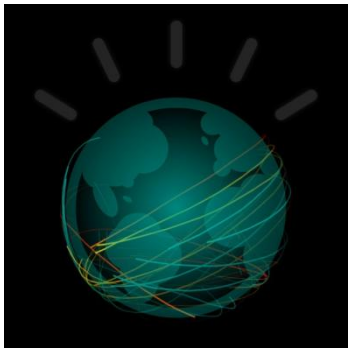
- 参赛者同时接收到一条线索(人类通过视觉/Watson以电子方式接收).
- 在灯亮后，玩家才可以应答.
- 人类可以预测; 而Watson则不能.

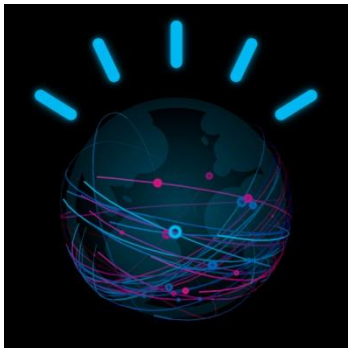


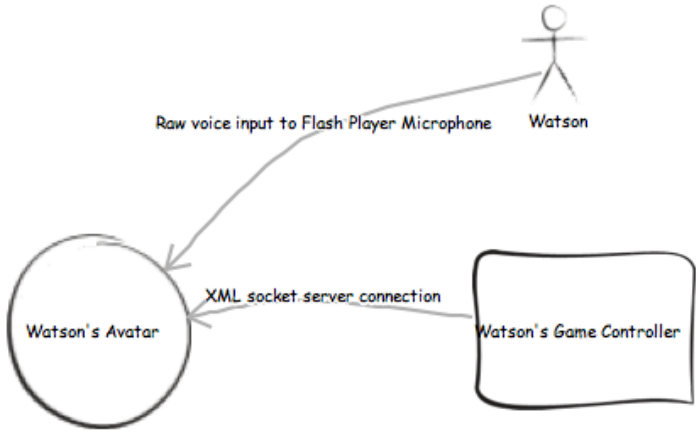
Deployment View: Avatar(The face of Watson)

- Developed by Joahua Davis and Branden Hall of **Automata Studios**.
- Designed using Adobe Illustrator CS5 and scripted with Adobe Flash Professional CS5 using the HYPE visual framework.
- Deployed using Adobe Flash Player 10.1.





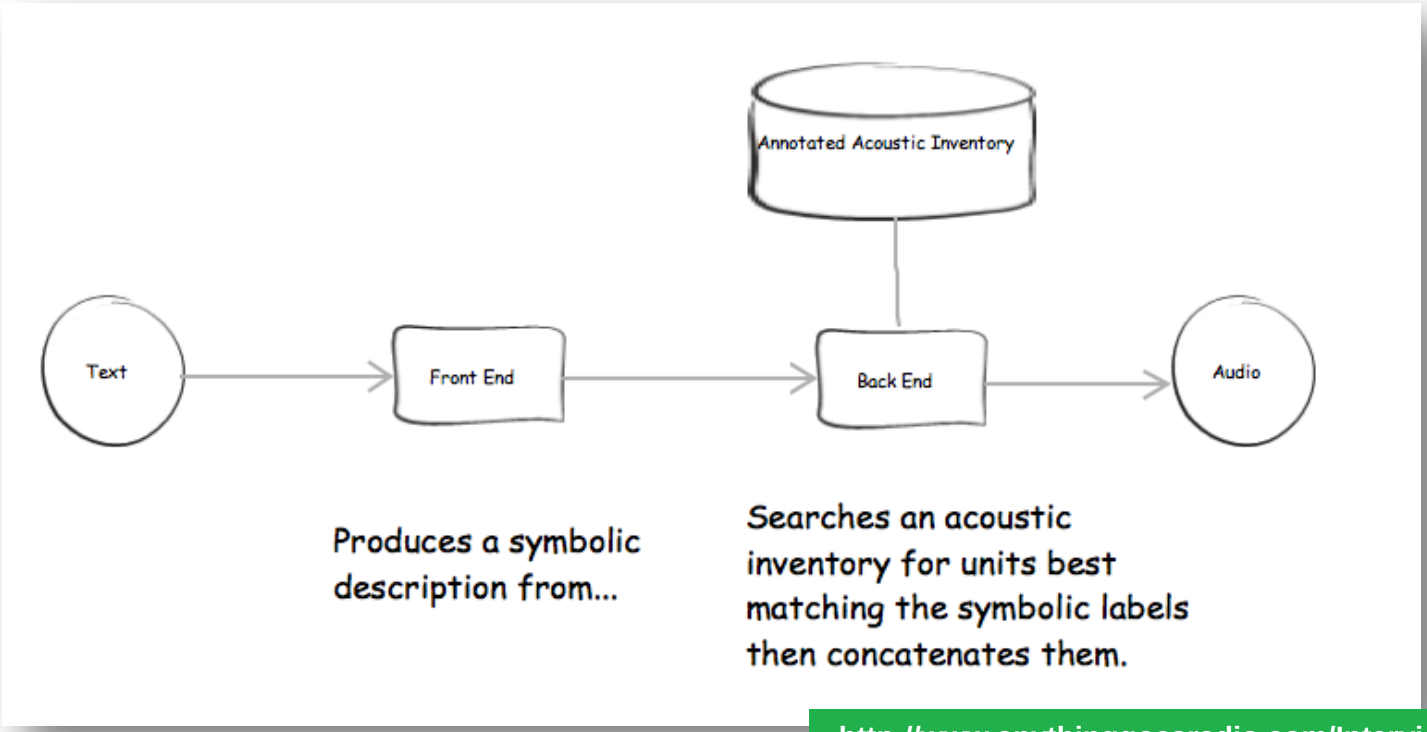




http://www.hypeframework.org/blog/content/ibm-watson-and-the-jeopardy-challenge/

Deployment View: Voice

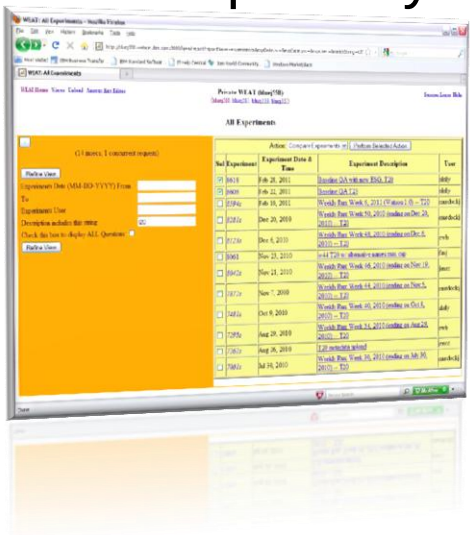
- 必须处理大量开放式的有相当难度的词汇表.
- 从Jeff Woodman的10小时音频中抽取音素做成文本到语音的合成素材.
- 前端是基于规则的语言学分析.
- 后端是一系列的韵律和声学模型.

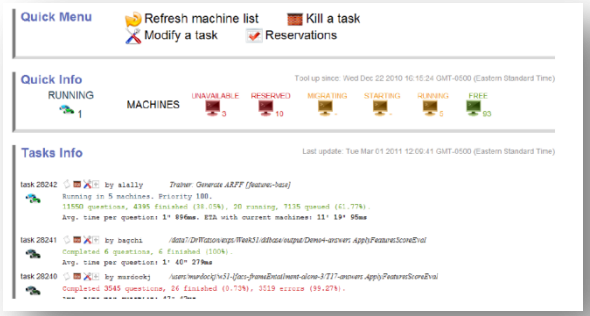


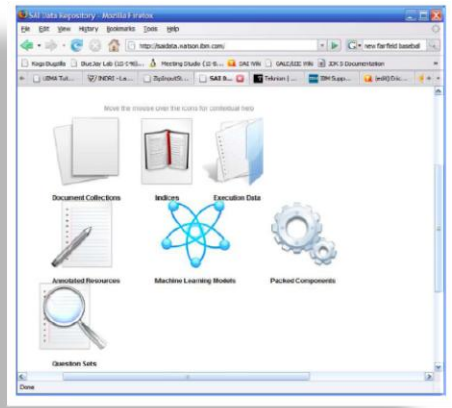
http://www.anythinggoesradio.com/Interviews/JeffWoodman_02_14_11.MP3

Tools

- Eclipse.
- Subversion -> RTC.
- Watson Error Analysis Tool (WEAT查看Watson如何思考).
- Feature Analysis Tool (FAT).
- BlueJ Automatic Distributed Execution Environment tools (BAIDE)
- Data repository tools.

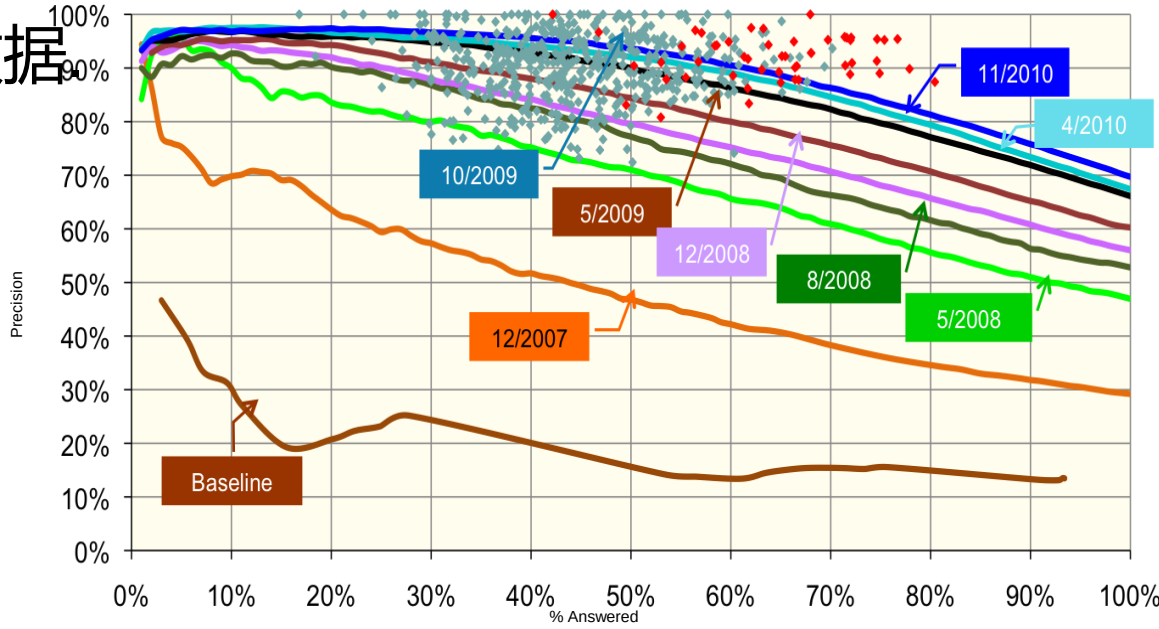




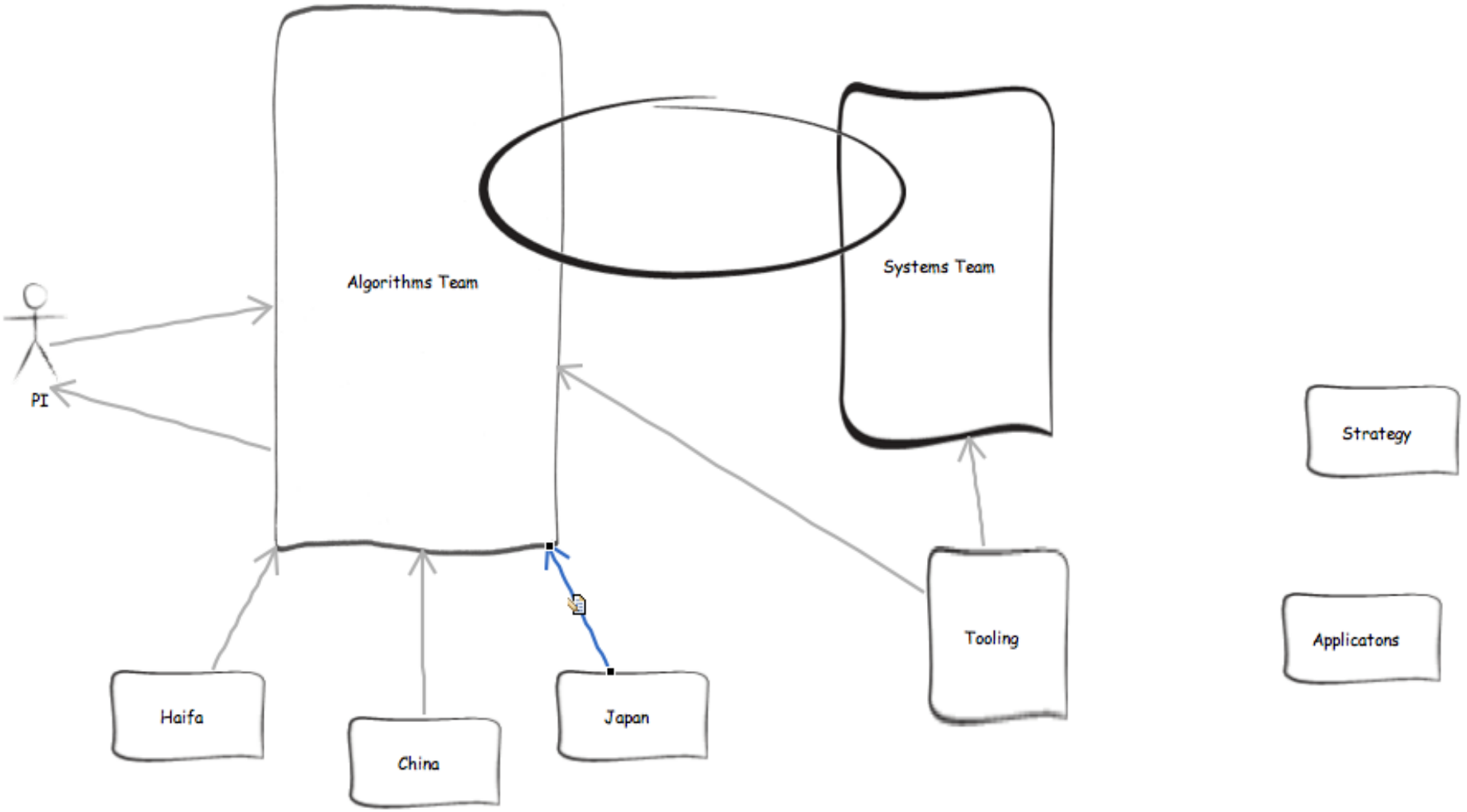


Process

- 敏捷开发.
- 持续协同的**War Room**.
- 每周集成.
- 结果驱动，端到端的集成测试.
- ~ 6,000 次实验
- 每周10 gigabits 测试数据



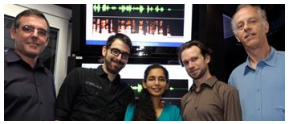
Organization



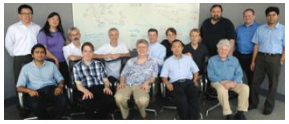
Organization

- **David Ferrucci**

- **语音组**



- **算法组**
  负责各NLP模块和其它模块的算法

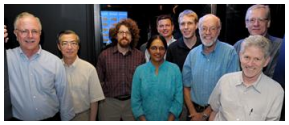


- **注释组**
  标注语料库，扩展机器人理解范围

- **策略组**
  处理流程和处理策略

- **项目管理**

- **系统组**
  处理流程和处理策略



- **中国研究院**
  本体、元数据、语义网络和知识表示

- **东京研究院**
  文本挖掘、文本抓取、半结构化信息使用

- **海法研究院**
  信息检索算法

- **应用组**

Deployment View: Watson构建于开放式标准、开源软件框架和标准化硬件体系

IBM DeepQA

Apache UIMA

Database/Wikipedia

IBM Power 7 Hardware

Used by Watson

IBM Content Analytics
Natural Language Processing and content analysis leveraging UIMA
Cognos, SPSS

InfoSphere BigInsights
"Big Data" analysis (**Hadoop**)

IBM Power Systems
Thousands of parallel processes

Workload Optimized Systems
Integrated, Optimized by Workload

Related Innovations

InfoSphere Warehouse DB2, Informix, Netezza
Aggregating and storing data and content

InfoSphere Streams
Massively parallel analysis

Business Analytics
BI, Predictive Analytics and more

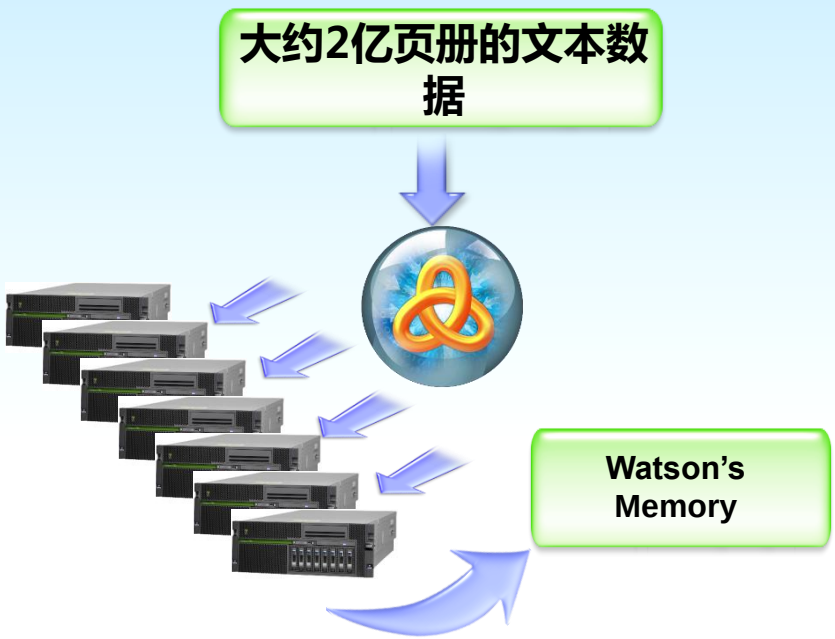
ECM Solutions
IBM eDiscovery Analyzer
IBM Classification Module
IBM OmniFind Enterprise Search

IBM Global Business Services
Research, expertise and analytical assets

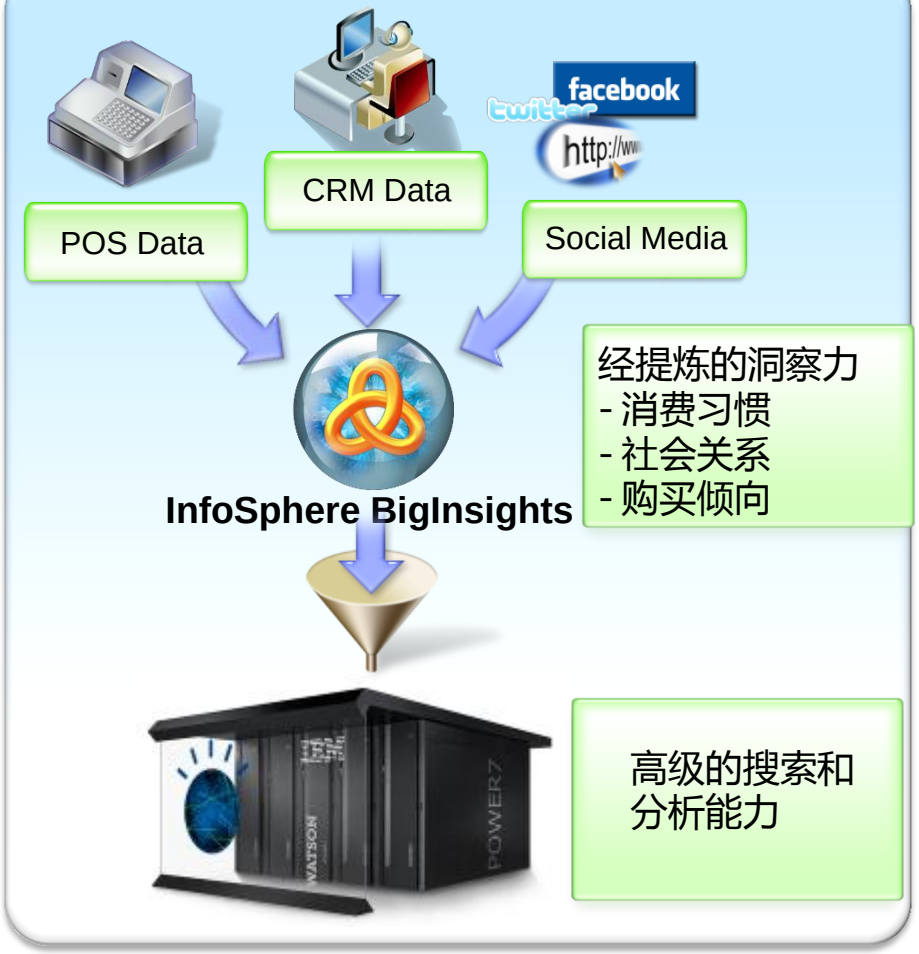
关键技术决策：商用Hadoop是Watson的信息处理基石

基于大数据的技术(BigInsights)构建了Watson的知识库

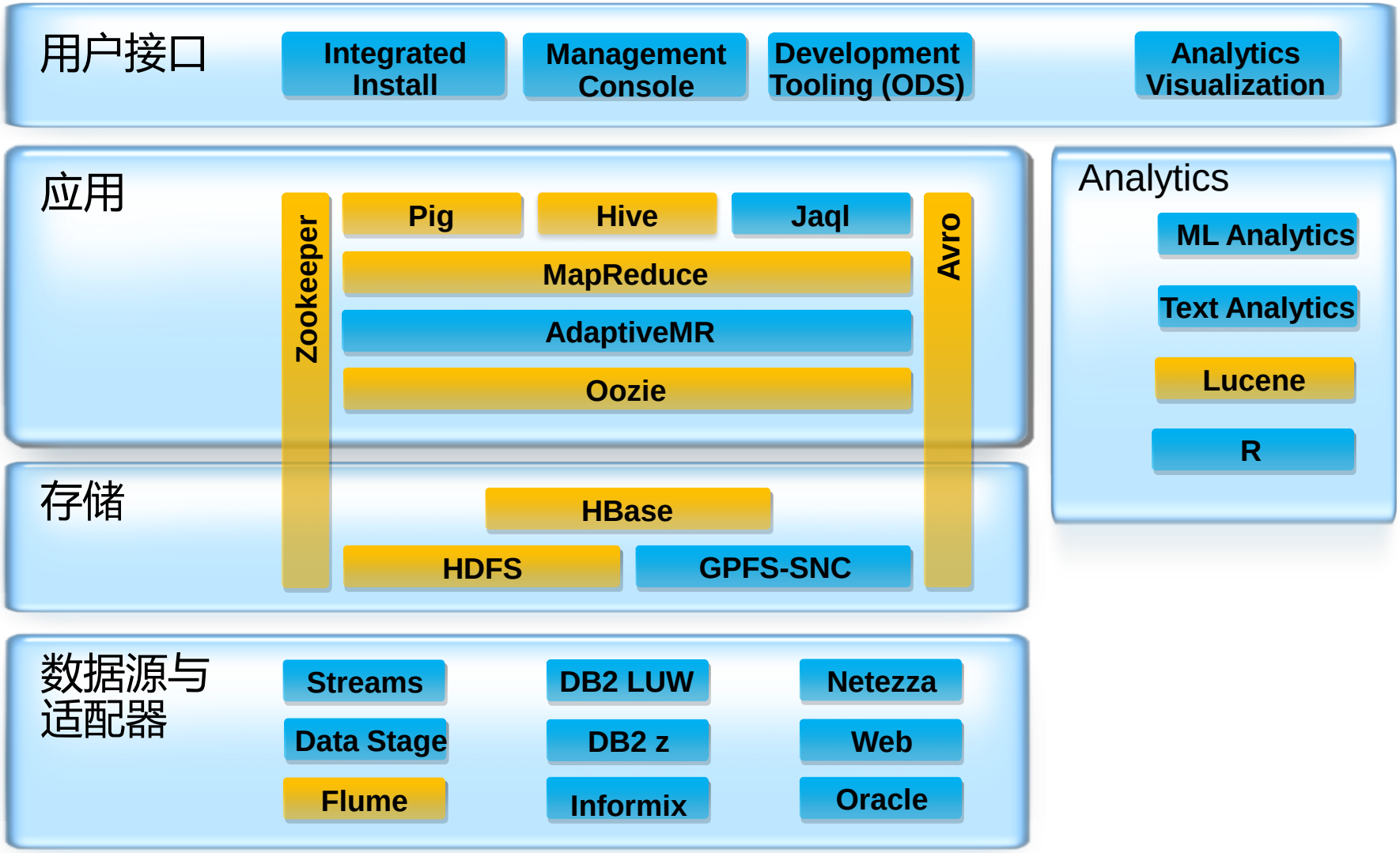
Watson使用Apache Hadoop 开源框架并行处理海量信息和工作量



Watson使用Big Data技术和产品为商业智能分析提供了绝佳利器

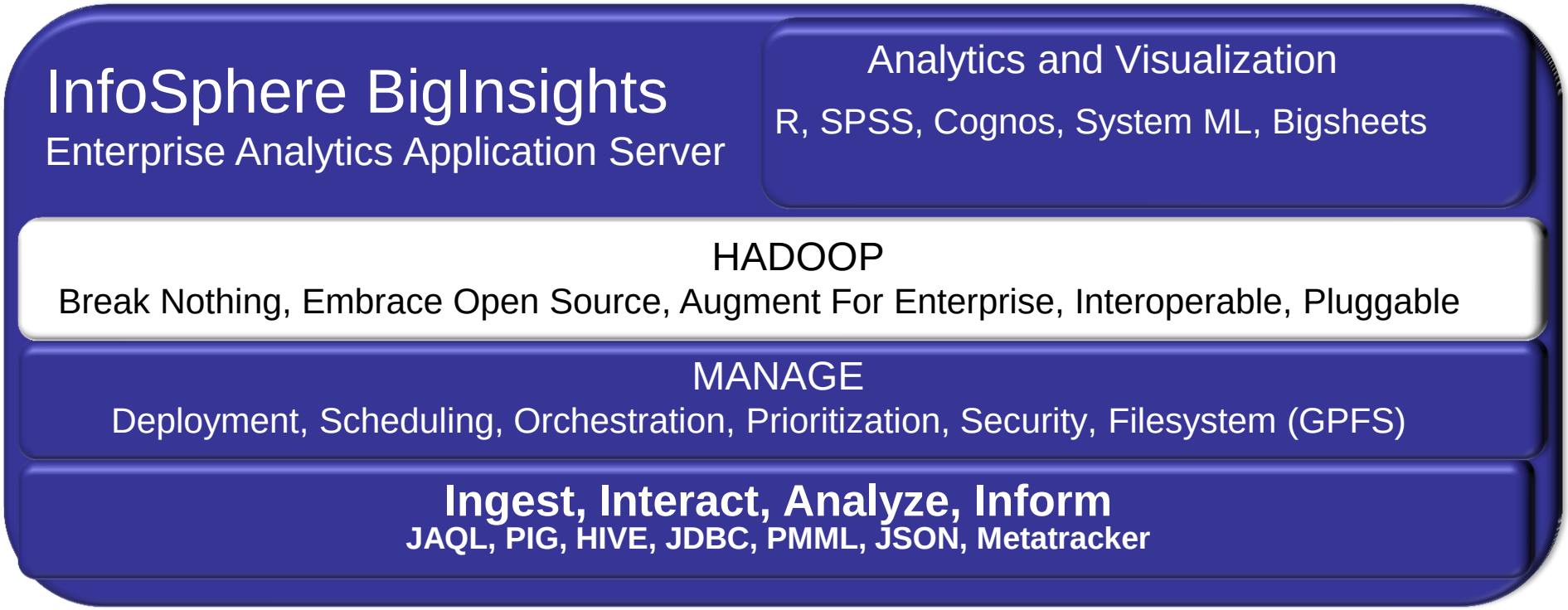


核心产品支撑：IBM BigInsights架构



IBM BigInsights – Hadoop made ready for Enterprise

- 通过客户协作构建创新
- 利用大数据分析加速创新平台的价值
- 提高性能、降低成本、降低风险



ETL

Data Warehouse

BI Platform

Enterprise Tooling and Metadata

实现决策：BigInsignts的优势

- **优势1：Internet级别的海量存储和企业级的高可用高可靠性**

  底层的hadoop技术是一个被证实的可以扩展到海量数据分布存储的分布式方案，解决了海量数据(半结构化和非结构化)的存储和访问问题，并且该方案是可以架构在低成本的机器集群上，提高了性价比。 BigInsights 底层存储GPFS-SNC基于GPFS发展而来，其借鉴了HDFS的一些设计理念，与HDFS相比，GPFS-SNC在性能、可靠性方面具有巨大优势，消除了HDFS的单点故障问题。

- **优势2：开放性的接口和集成能力**

  InfoSphere BigInsights可以与IBM数据分析软件深度集成，可以提供更为强大的分析能力，基于Java和开源技术，不存在技术壁垒，方便用户自定义开发和客户化该平台。高度的集成性还体现在可于数据仓库中运行第三方的分析模型，并与分析应用和分析模型进行端对端集成，避免海量数据的加载等。这些开放的能力可以继续利用客户现有的分析平台的投资，降低整体的拥有成本。包括企业级别的数据仓库集成能力(Netezza, DB2，InfoSphere Warehouse)

- **优势3：企业级别管理和易用性增强，数据分析能力增强**

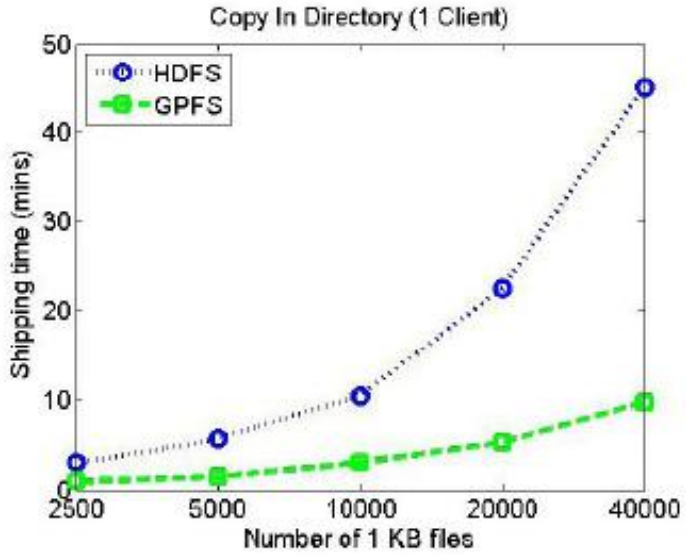
  - ✓ IBM实验室前端技术的结晶(文本，图像，视频分析)
  - ✓ 简单但是具有强大的扩展能力的JAQL语言
  - ✓ 统计分析平台project R 以及机器智能学习systemML
  - ✓ 可视化的工具展示挖掘
  - ✓ 高可用性和性能增强
  - ✓ 管理与易用性增强
  - ✓ 安全性增强
  - ✓ IBM服务支持

技术可行性分析

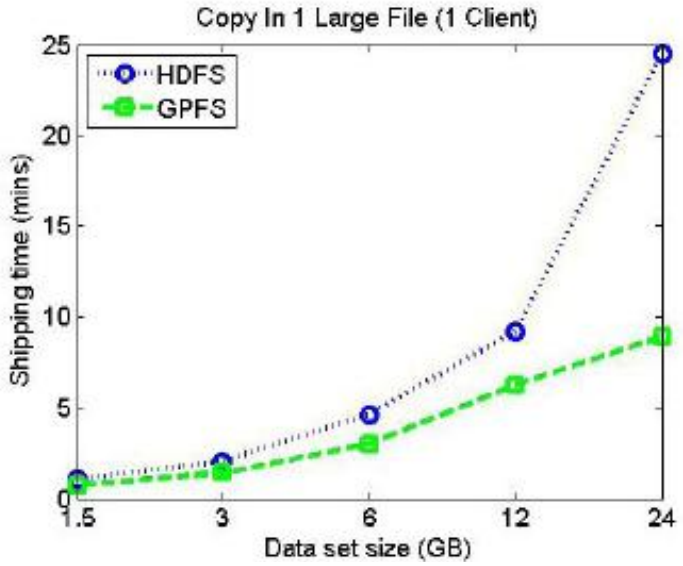
	IBM GPFS-SNC	开源HDFS或其他方案
健壮性	无单点故障 99.99%	NameNode 存在单点故障
数据一致性	高	数据可能会丢失
可扩展性	高，实测4000+	高，新版本不断发展中
POSIX 兼容	完全兼容	有限
数据管理能力	安全、备份、快照、缓存、广域网复制	有限
传统应用性能	好,兼顾读写性能	随机读写性能差
安全性	支持ACL, 容量限制，安全认证	不支持

GPFS-SNC与HDFS性能比较

Metadata Efficiency (1 KB files)



Data Efficiency



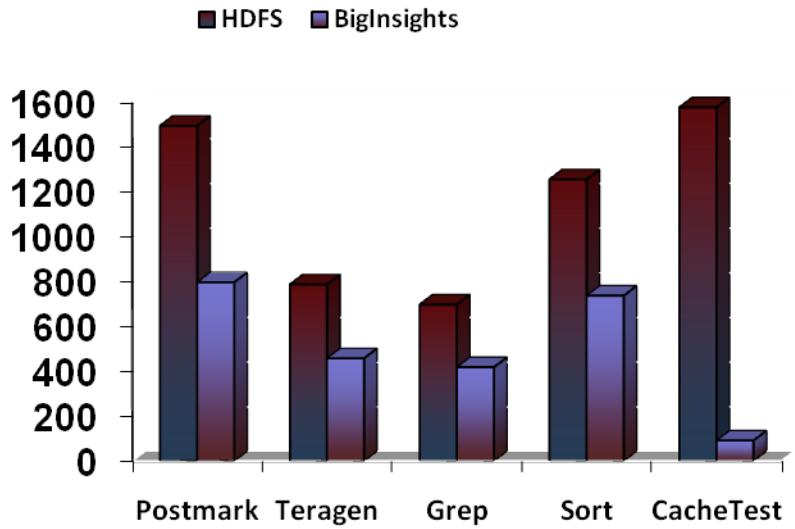
GPFS-SNC vs HDFS 效率对比

注 - GPFS-SNC: General Parallel File System-Shared Nothing Cluster

GPFS-SNC 整体性能提升

- 对于需要随机I/O的 SQL查询,JAQL查询等
- 对于需要线性排序的sort操作来说
  - BigInsights 提供2至3倍与开源Hadoop技术的性能
- 文件索引等查找效率
  - 17倍于开源Hadoop技术,得益于Client Cache能力

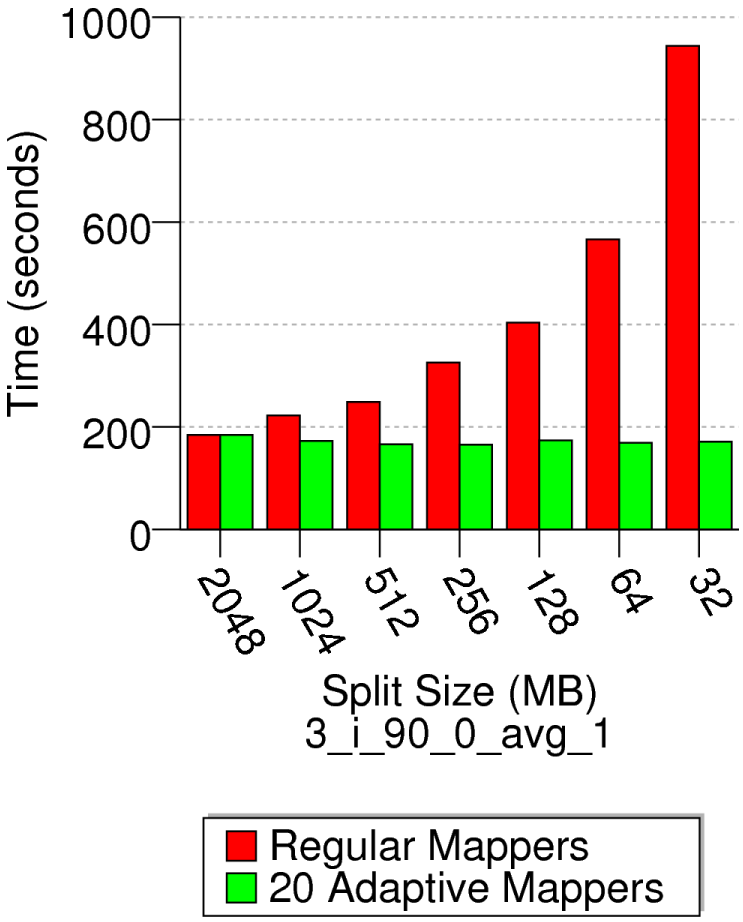
Test	BigInsights	HDFS
1 TB Terasort	6 hrs 19 mins	10 hrs 28 mins
16-node iDataPlex 2.2 GHz Xeon 2x4-core 8 GB RAM, 4x750GB SATA, Hadoop-0.20.2 4-disk RAID0 tmp space for better sort performance. Faster outer tracks of disks allocated to HDFS Data Integrity issues observed in HDFS		



AMR-Adaptive MapReduce

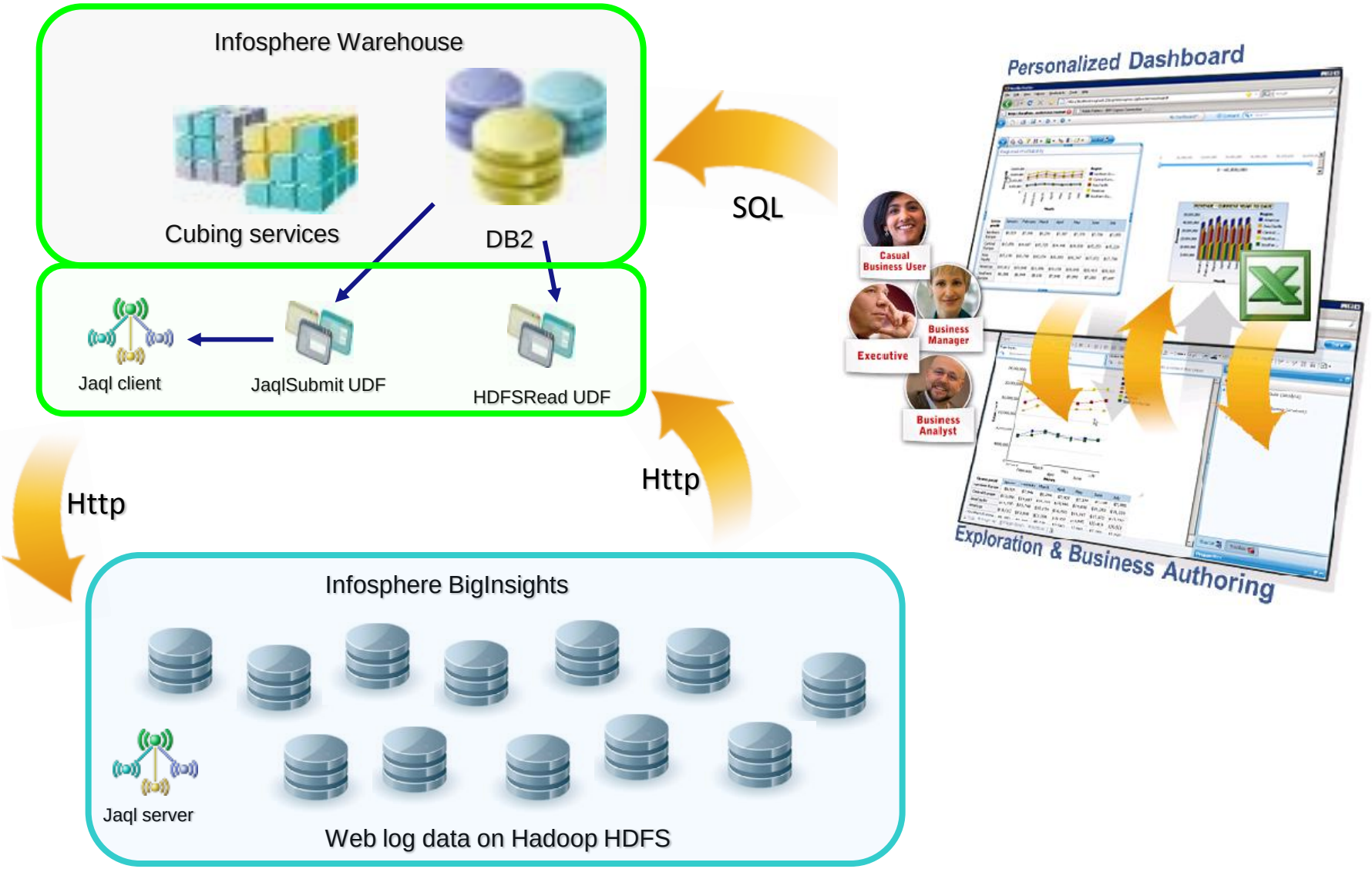
- 什么是AMR
  - IBM研究实验室发明的能够加速小MR任务处理速度的方法，对于创建任务来说透明，无需改变应用
- 为什么要用AMR
  - IBM研究表明，大多数的用户系统资源会被大量的小任务消耗殆尽，系统中30%的任务能够通过AMR得到性能提升
  - 大规模Hadoop集群中性能调优非常困难，很多参数对系统整体影响非常大，少的有2至3倍，多的能到10倍以上
  - 构建的MR代码不够优化
- 典型应用场景
  - 多用户的用户场景能够从AMR受益
  - 分析的任务中所要处理的数据类型和数据量比较小的场景中

AMR性能



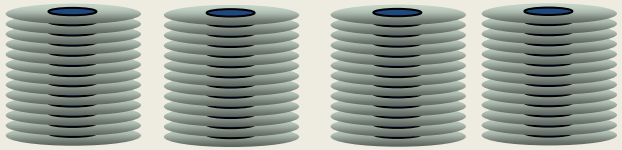
- 与Hadoop 任务全面兼容
- 测试表明有20%-50%的性能提升
- 在运行时自调整
- 以Broadcast join为例
  - 减小MR任务启动耗时
  - Map任务中的数据不平均
- AMR使得
  - Map中Split对于数据块大小不在敏感
  - 默认数据块(64MB)性能表现大幅提升

开放性接口和集成能力：连接DB、DWE和BI等

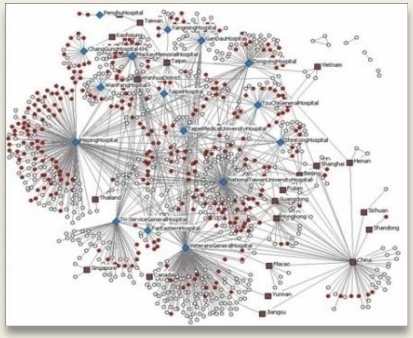


IBM Netezza 分析：大数据遇到大智慧

- 目标定制的分析引擎
- 集成的数据库、服务器和存储



- 标准接口
- 总体拥有成本低



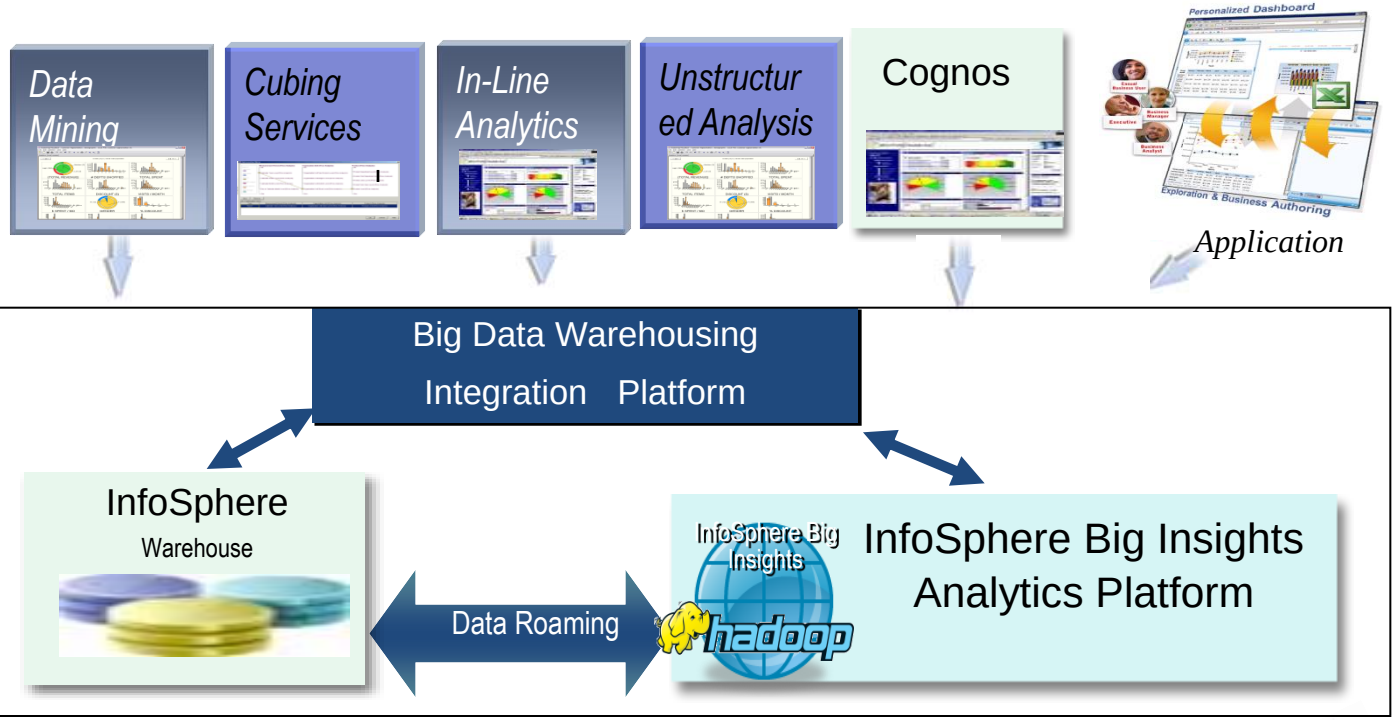
- 快速：比传统系统快
- 简单性：管理和优化操作最少

- 可扩展性：PB 级用户数据容量
- 智能：高性能高级分析

无约束分析

混合模式的信息管理

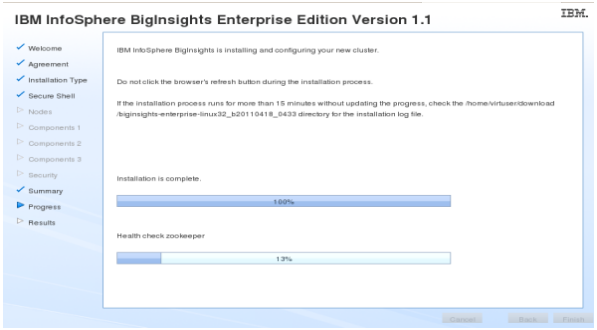
- 用较经济的容易扩展的的Hadoop系统扩展分布数据服务
- 实时接入到大数据集合和 RDBMS
- 统一的 BIaaS, DaaS
- Big Insights与传统数据库无缝集成



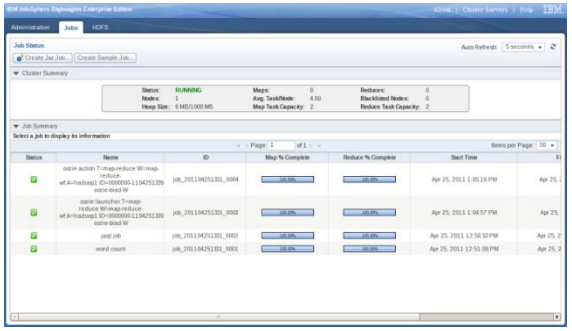
集成的安装和基于Web的管理控制平台

- 集成的基于Web的安装
  - 无缝的单节点或者集群模式安装
  - 开源组件和IBM组件的安装验证检查，确保系统正常运行

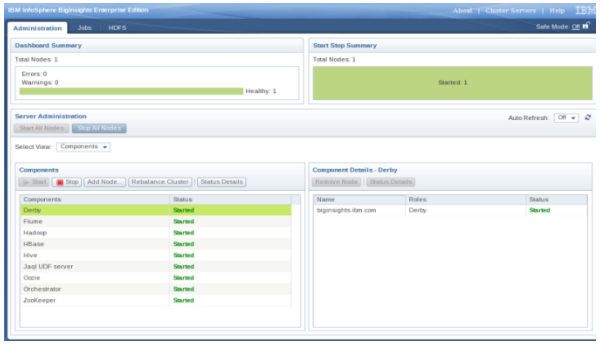
- 基于Web的管理控制台



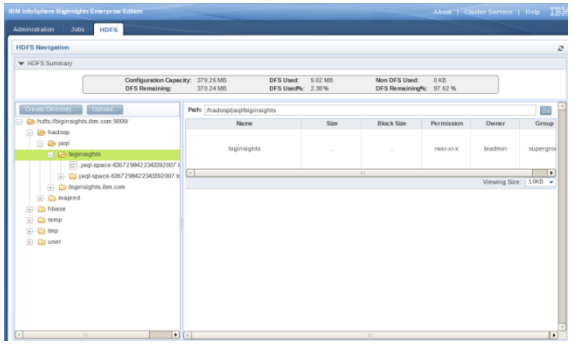
集成安装



任务和工作流管理



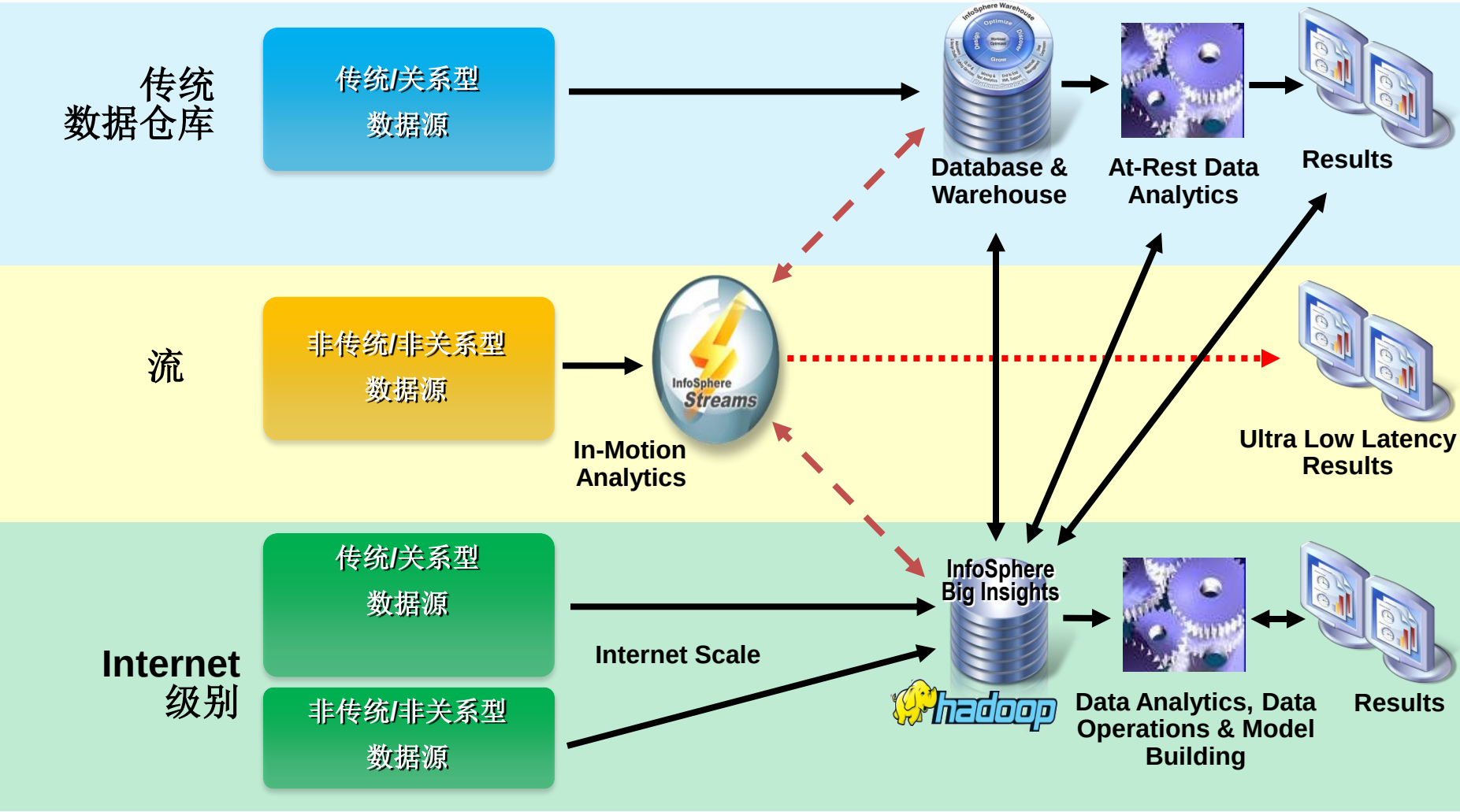
系统健康监控



集群以及文件系统管理

IBM 的端到端，整体解决方案

超越传统的数据仓库概念



未来

預祝2012
新年進步，架構功成！

艾飛 @AlexPitt
aifei_bj@yahoo.com.cn