Continuity of Veteran Care during EHR Modernization and Beyond

VISTA Data Project

VHA Health Solutions Management Brief

July 24, 2017

Rafael Richards MD MS
Director, VISTA Data Project
Veterans Health Administration
U.S. Department of Veterans Affairs

An interagency project with the U.S. Department of Defense, Defense Health Agency

- VA-DoD Interagency Project
- EHR modernization Proof of Concept
- Leverages DoD-developed technology
- Formalizes Veterans Care Model
- Execution 2016-2017
- http://vistadataproject.info

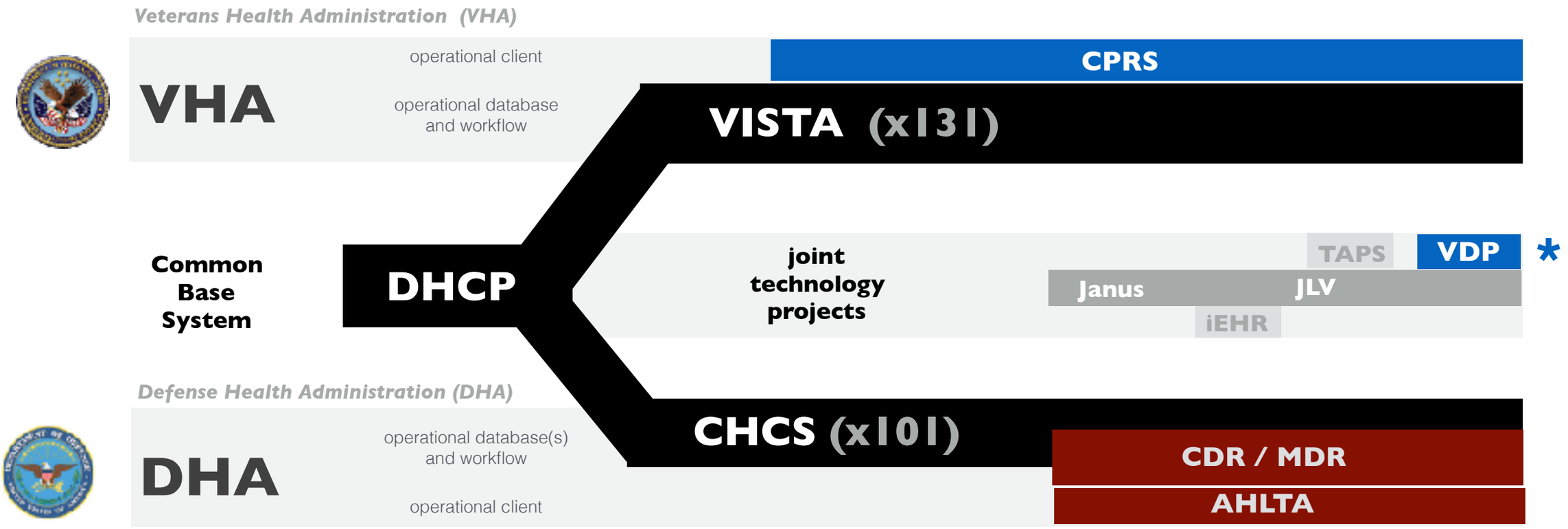
VHA-DHA: History of Electronic Health Records

DHCP is the common base system

VHA: 151 hospitals; 820 clinics; 300 vet centers; + other (total 1700 care sites)
DHA: 57 hospitals; 350 clinics + other

VHA: 131 VISTA systems operational (since 1981)
DHA: 101 CHCS systems operational (since 1985)
Total: 232 DHCP-based systems across VHA-DHA

DHCP-based systems
Common technology projects
VHA-specific interface and workflow
DHA-specific interface and workflow



While DHCP was similar in VHA and DHA originally, it has diverged over time. Today the variety and volume of CHCS data is approximately one-third the scope of VISTA data. One reason for the difference is that DHA migrated a large portion of CHCS operational data and functions to CDR / AHLTA.

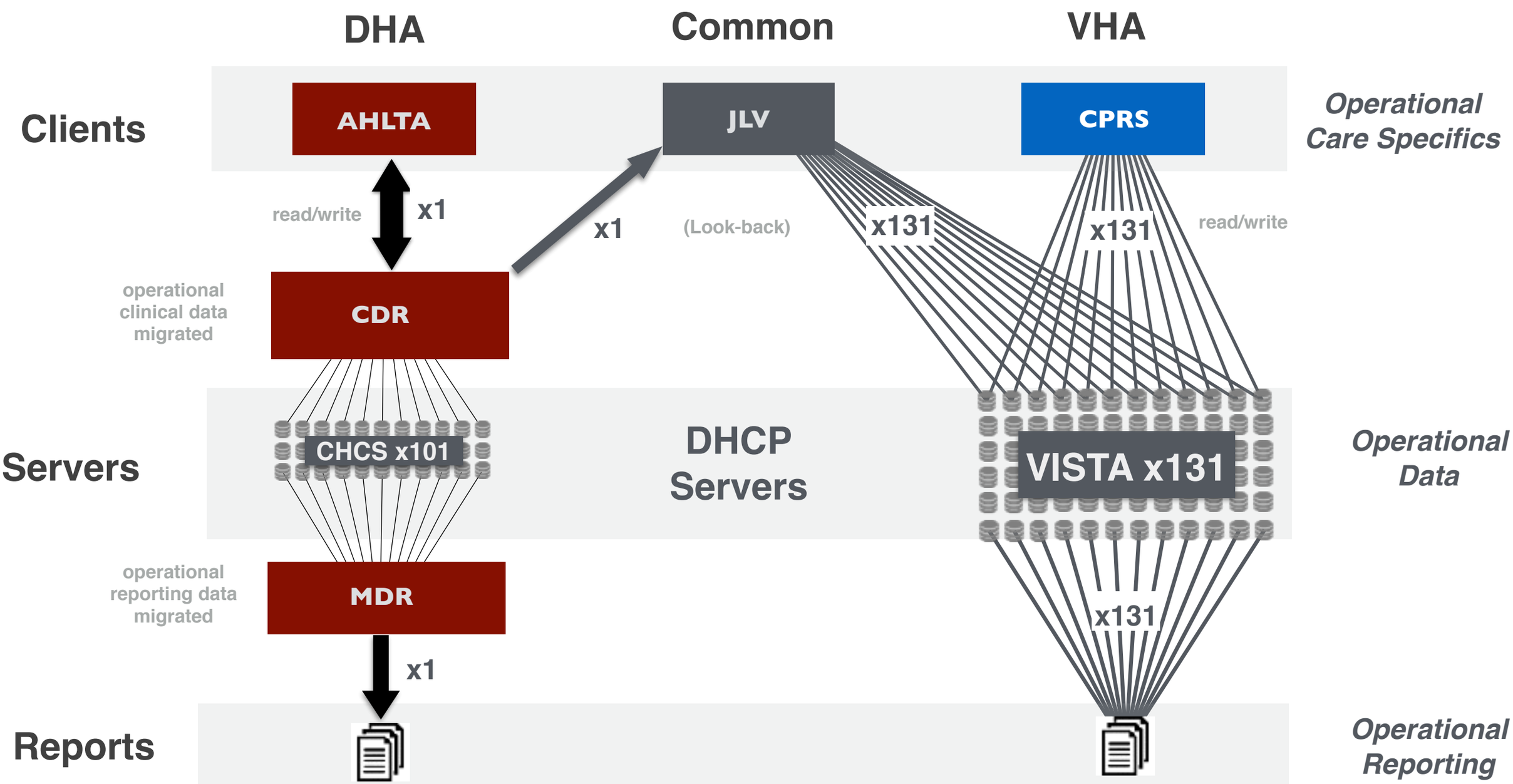
Note: Time scale simplified for clarity

	1980	1990	2000	2010	present
VHA-specific		**VISTA**	CPRS		
Common	**DHCP**		JLV	iEHR TAPS	VDP
DHA-specific		**CHCS**	AHLTA / CDR		Genesis

1981 - DHCP - Decentralized Hospital Care Program - VA Fileman database and applications [VHA]
1985 - CHCS - (DHCP renamed) Composite Health Care System; modified for DHA use [Leidos (SAIC)]
1994 - VISTA - (DHCP renamed) Veterans Information Systems Technology Architecture [VHA]
1997 - CPRS - Computerized Patient Record System - graphical interface and workflow [VHA]
2004 - AHLTA/ CDR/ MDR - Armed Forces Health Longitudinal Technology Application [Northrup Grumman]

2003 - JLV - (originally Janus; renamed to JLV in 2011) [DHA-VHA]
2011 - iEHR - Integrated Electronic Health Record [SMS]
2013 - TAPS - Transition Application Plan Support [DHA-VHA]
* **2016 - VDP - VISTA Data Project [DHA-VHA]**

VHA-DHA: DHCP Servers today



DHA migrated much of its operational data from CHCS to MDR / CDR .

Today the variety and volume of CHCS data is approximately one-third the scope of VISTA data. One reason is that DHA migrated a large portion of CHCS operational data and functionality to CDR / MDR.

VHA has not decided on its long-term strategy for operational data.

CPRS: Blueprint for Veteran Longitudinal Care

CPRS is VISTA to Physicians, and Embodies Veteran Care specifics

Veteran-specific

Built specifically around veteran care policies and practice

Department of Veterans Affairs **Memorandum**

Date: OCT 17 2002

From: Deputy Under Secretary for Health for Operations and Management (10N)

Subj: National Patient Record Flag for High Risk for Suicide

To: Network Director (10N1-23)
Chief Medical Officer (10N1-23)
Network Mental Health Liaisons

1. The purpose of this memo is to provide guidance for the implementation of a new Category I Patient Record Flag (PRF) for High Risk for Suicide.

Agent Orange

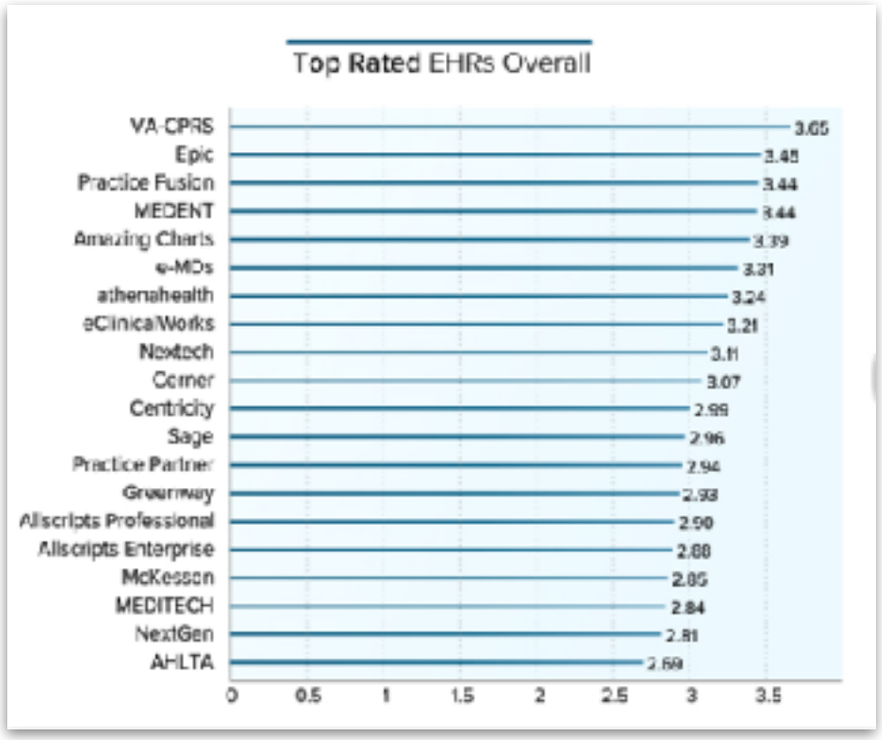
Agent Orange (AO) is an herbicide that was used in Vietnam between 1962 and 1971 to remove unwanted plant life that provided cover for enemy forces. The VA has recognized the following conditions as associated with but not necessarily caused by exposure to Agent Orange:

- AL Amyloidosis
- Diabetes (type 2)

Physicians favorite

Medscape EHR Report 2016: Physicians Rate Top EHRs

Carol Peckham, Author; Leslie Kane, Sr. Director, Medscape Business of Medicine; Susanna Rosensteel, Editor | *August 25, 2016*



http://www.medscape.com/features/slideshow/public/ehr2016

Opportunity:
Supporting CPRS (for a period) ensures Continuity of Care as VA's EHR is modernized.

CPRS workflow: Captured in VISTA Server

CLIENTS
(User Interfaces)

Terminal Interface
(integrated in VISTA server)

Graphical Interface
(remote from VISTA server)

1981 **1997** **2017**

INTEGRATED SERVER:

Database + Applications + Workflow Engine + Integration Engine + Terminal Interface

ScreenMan

RPCs (workflow)

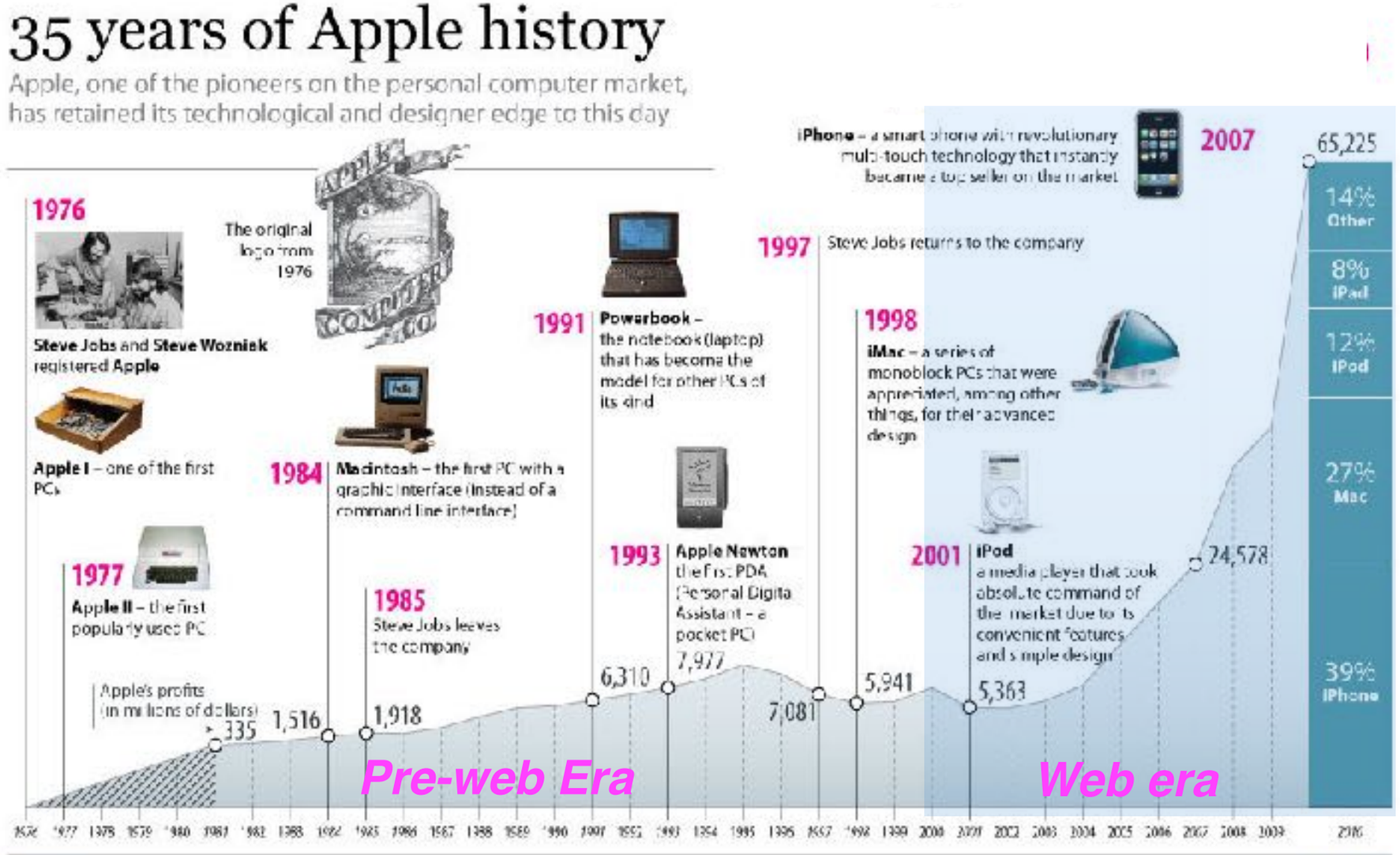
VISTA Server (x131)

1980 1990 2000 2010 present

Because the VISTA server's remote procedure call (RPC) interfaces captures the clinical operational data and workflow of the CPRS client, migrating CPRS/VISTA "server-first" ensures VA Continuity of Care.

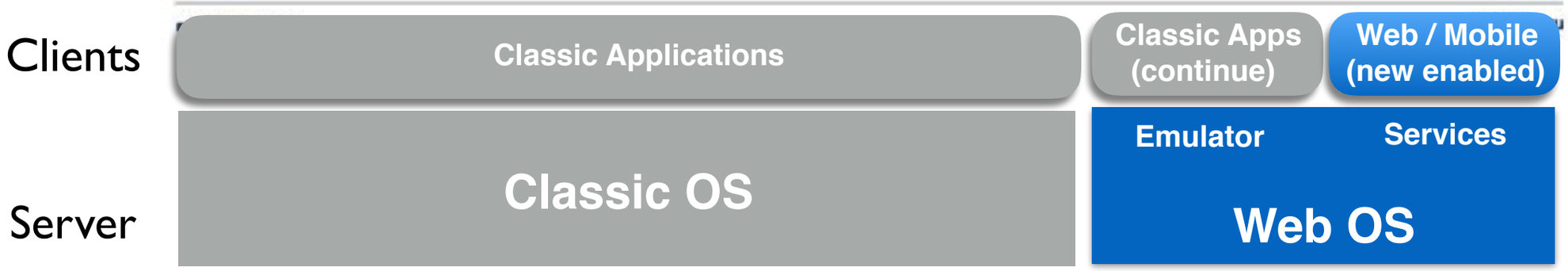
Apple: Playbook for platform evolution

Emulation was their key to platform modernization



In 2001 Apple migrated from their bespoke pre-web "Apple Classic" operating system to a to a modern, web-centric OS.

This new web OS allowed Apple to expand their platform, services, and applications across all devices, and into the Cloud, giving them exponential growth and market reach.



Apple's seamless migration to a modern web OS while maintaining continuity of operation of their Classic applications was enabled through Emulation.

VISTA Data Project
https://vistadataproject.info

VISTA Data Project

Prove Stepwise *Server Migration*
while maintaining Continuity of Care

Key Features

- VHA-DHA Interagency project
- Modernization Proof of Concept
- Leverages DHA-developed technology
- Formalizes Veterans Care Model
- Migrate Server; Support CPRS/JLV
- Execution 2016-2017

131 Current VISTAs

CPRS / JLV ….

RPC Interfaces (x 1000s)

or

Insecure Asymmetric Opaque Legacy MUMPS Code-driven Interfaces

Current VISTA Server

RPC Interfaces (includes VPR)

VISTA Applications

VA Fileman Database

MUMPS-driven VISTA Server
(maintenance, growth, and usability issues)

Stepwise, Measurable Migration

Time

Pass Thru

Pass Thru | Emul

Pass Thru | Emulate

Pass | Emulate

Emulate/REST

VISTA Data Project

CPRS / JLV (current continue)

web/mobile (new enabled)

or

Secure Symmetric Modern Node.js Model-driven Interface

RPC Emulator | VPR Emulator | REST

Clinical Domain Services

Master VISTA Data Model

VISTA Data Model

VA Fileman Database

Node

node VISTA Server

Structured VISTA Server
(mainstream, modular, extensible)

M Legacy VISTA (MUMPS)

Master VISTA Data Model (MVDM) Node.js - Driven VISTA

VISTA Data Project

Strategic Benefits

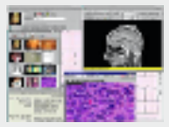
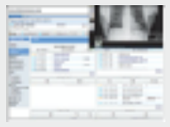
- *New, maintainable veteran care **server** based on mainstream technology*
- *New web and mobile clients enabled with mainstream technology*
- *Current clients (CPRS/JLV) supported and enforce VA Care coverage*
- *May now safely incrementally retire legacy MUMPS VISTA [spaghetti]*
- *(Some) Clinical Domain Services may be implemented over COTS*

VISTA Data Project

Stepwise Server Migration while maintaining Continuity of Care

Enables Cloud-based, COTS-integrated
National Veteran Care Services
Preserving Continuity of Care

Supports the current…


CPRS *JLV* *VPR*

Enables the new…

web *mobile*

commercial

Clients

Current and New Clients

Security Enhanced

Veteran Care Specifics maintained

Secure Symmetric Modern Node.js Model-driven Interface

Servers

- RPC Emulator | VPR Emulator | REST
- Clinical Domain Services
- Master VISTA Data Model
- VISTA Data Model
- VA Fileman Database

Node

node VISTA

New, Structured VISTA Server
(mainstream, modular, extensible)

Resources

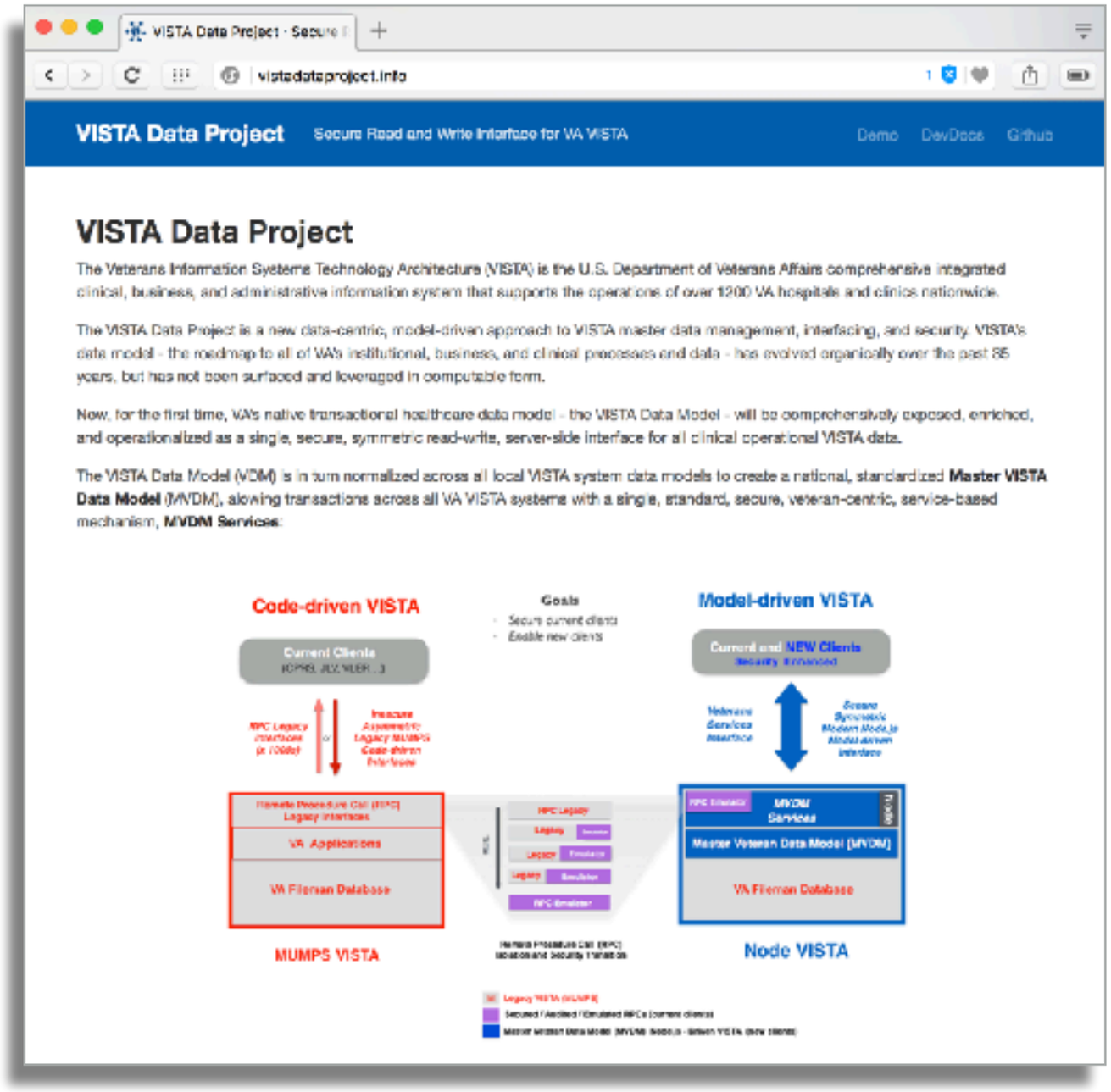
Web: vistadataproject.info

Github: github.com/vistadataproject

Contact: rafael.richards@va.gov

VDP Website

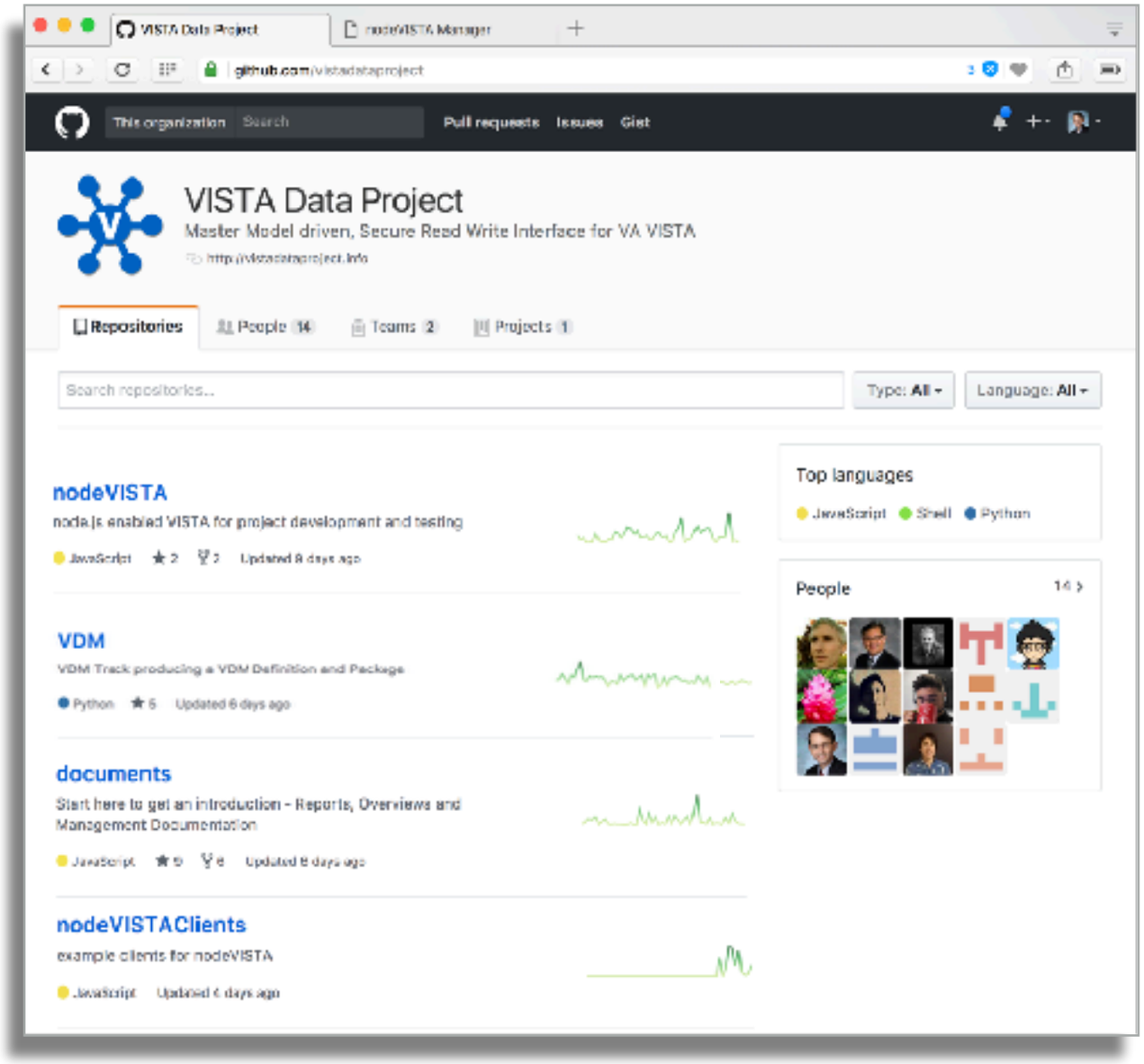
Home page and documentation



http://vistadataproject.info

VDP Github

All project code, artifacts, and management



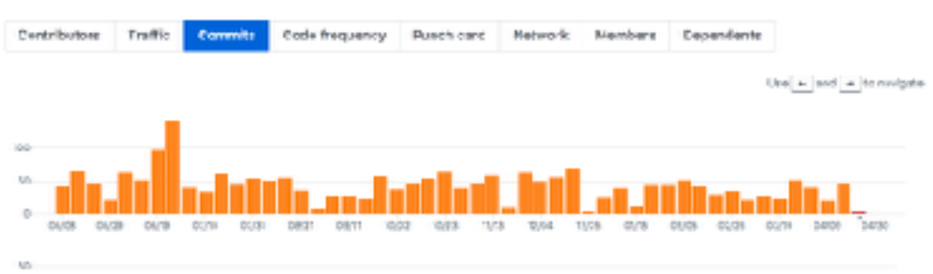
https://github.com/vistadataproject

VDP Github

Real-time federal interagency coordination, development, management, and analytics

Real-time cloud-based collaborative development, documentation, and management distributed across two federal agencies and developers across ten time zones.

Real-time team and progress tracking



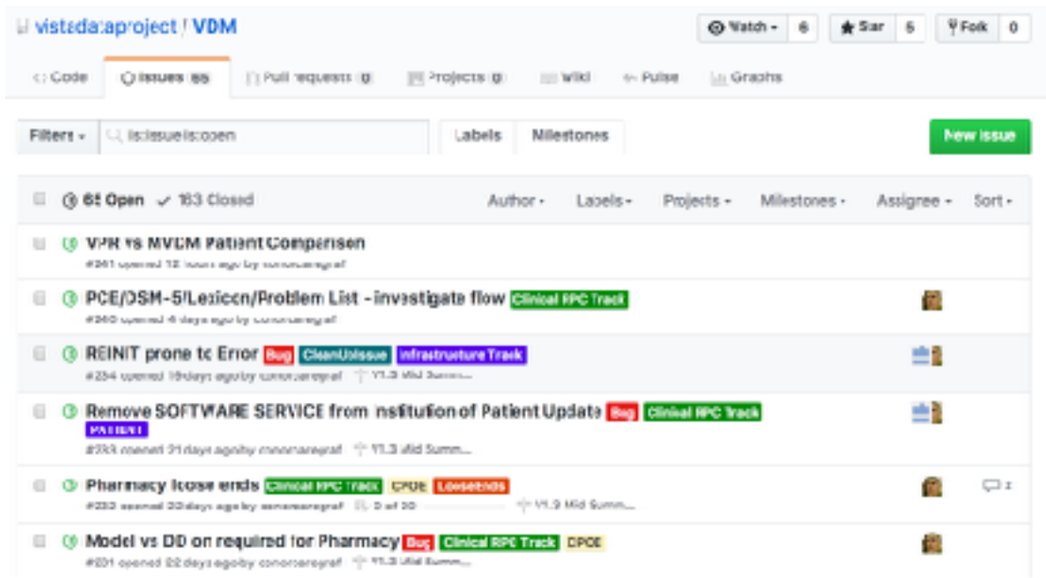
Collaborative cloud-based documentation

Welcome to the VDM wiki!

Developer How tos/Abouts

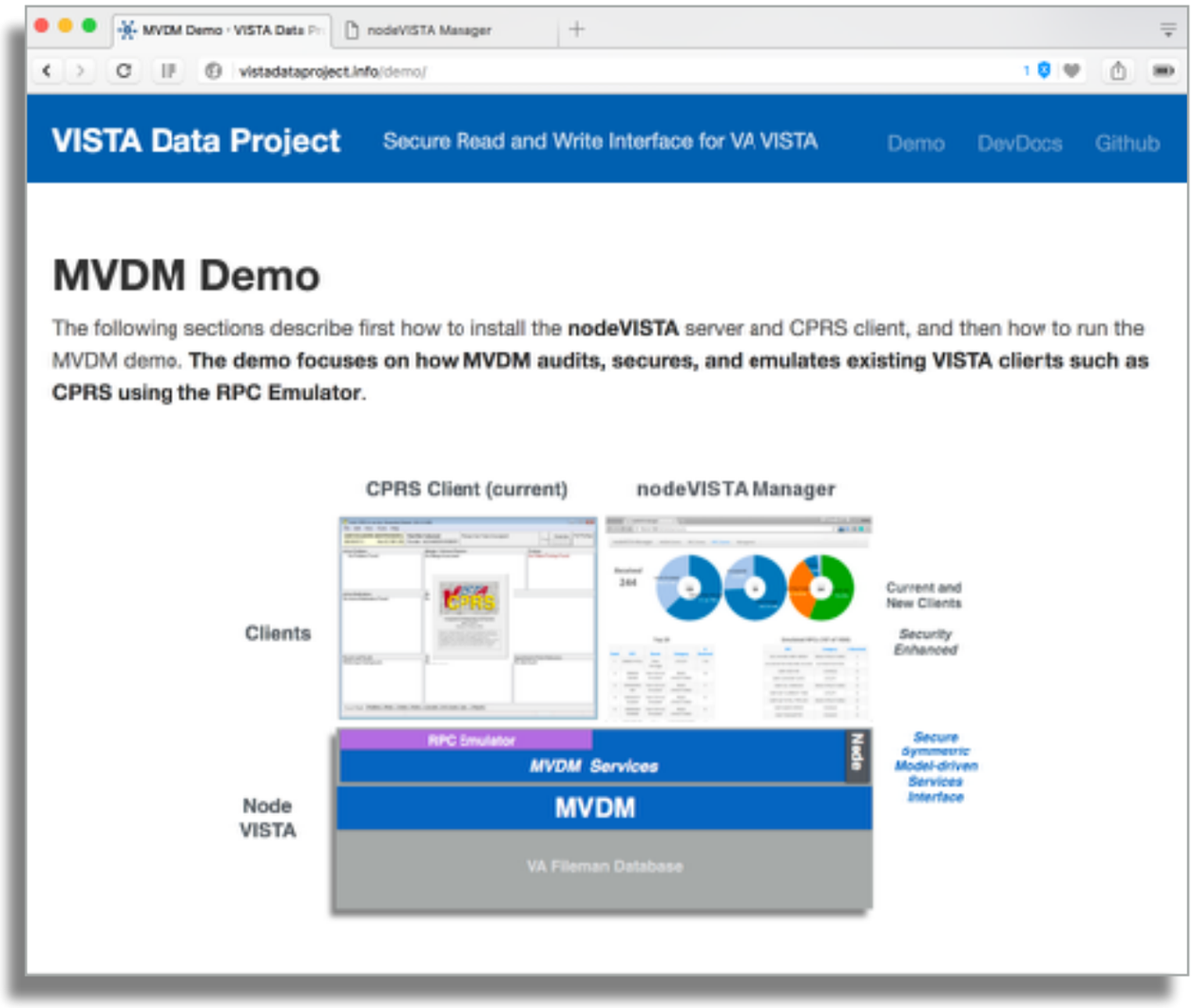
- Server Development Environment FAQs
- Running FileMan delta
- Debugging via node-inspector on nodeVISTA
- Running CPRS with debug
- A Native VISTA Data Model (VDM) over FileMan (Backgrounder)
- Locking RPCs
- Extracting MVDM from VDM - MVDM Git created from VDM

Real-time issue tracking and remediation



VDP Github

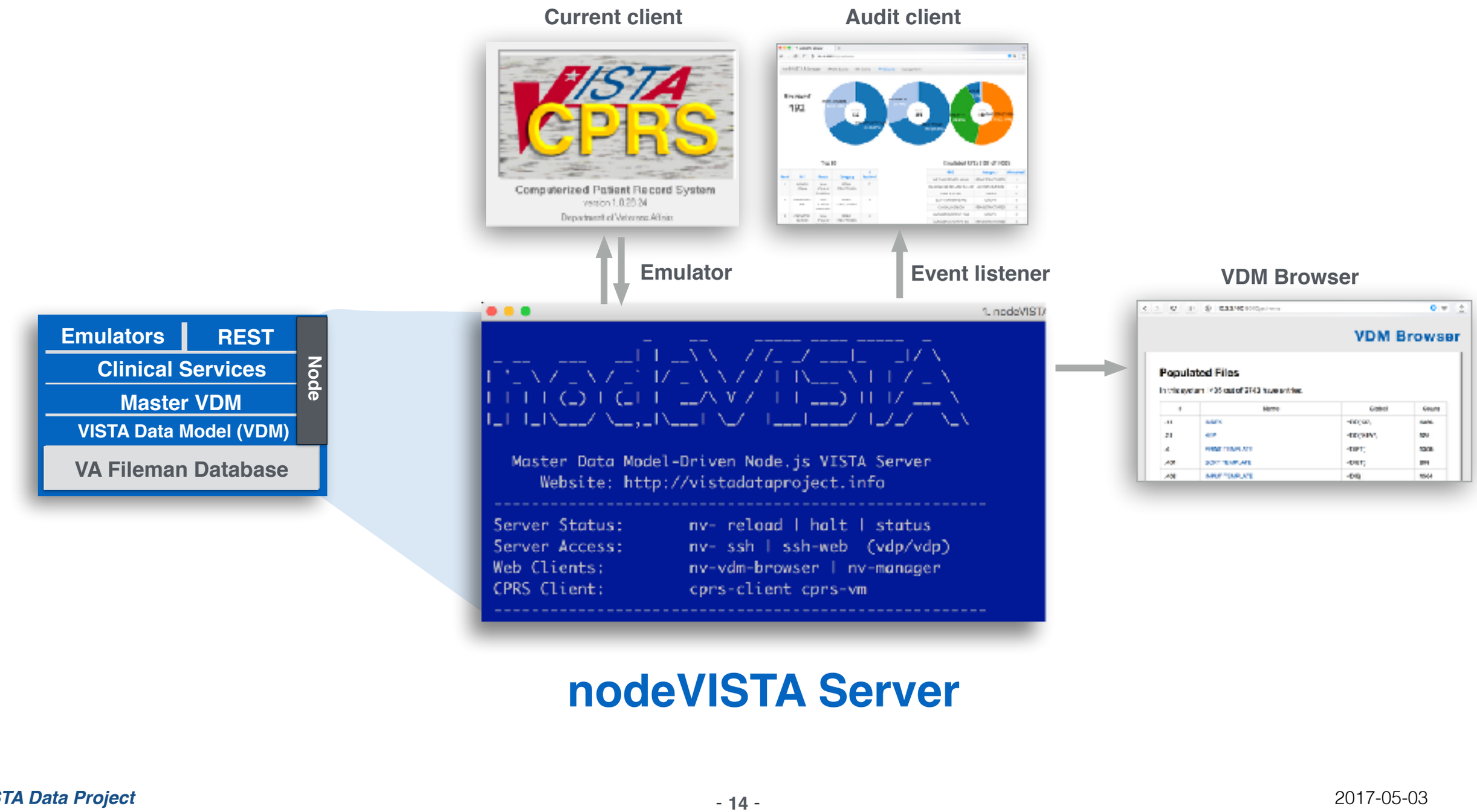
nodeVISTA downland and demo



http://vistadataproject.info/demo

VDP: nodeVISTA

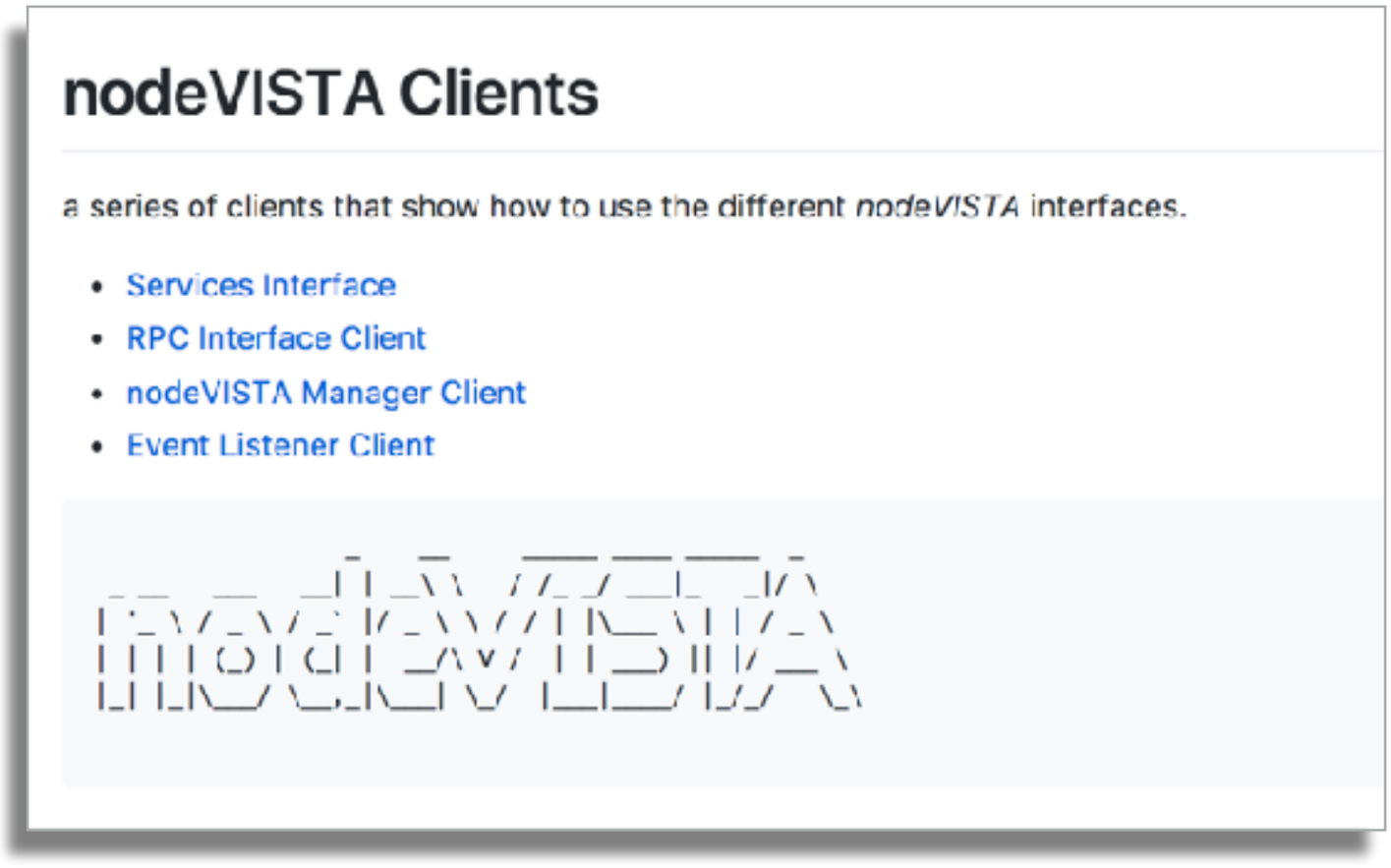
nodeVISTA downland and demo



nodeVISTA Server

VDP Github

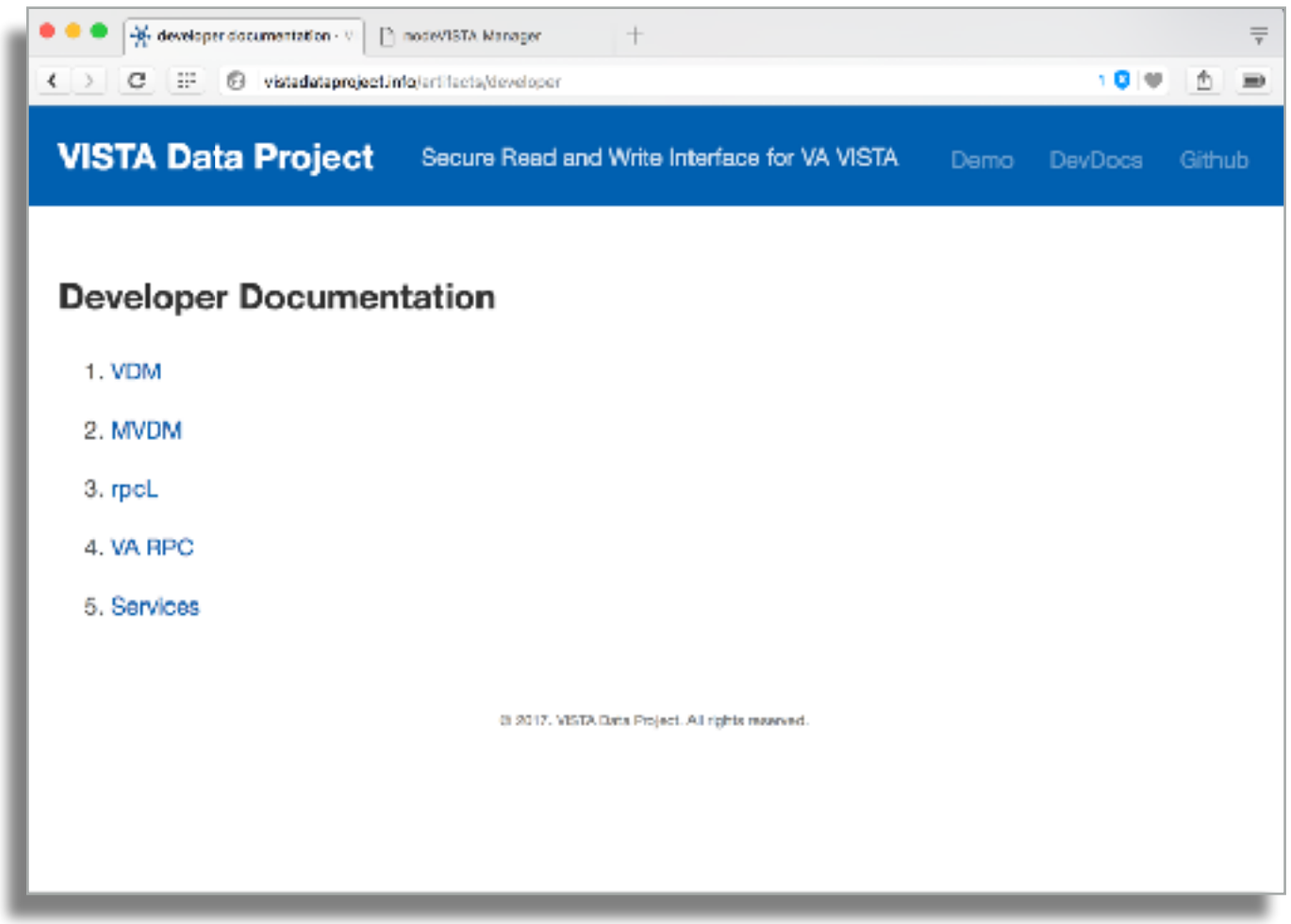
nodeVISTA Clients



https://github.com/vistadataproject/nodeVISTAClients

VDP Github

Developer documentation



http://vistadataproject.info/artifacts/developer

Technical

VDP Github

Standards and Technologies

Standards

Technologies

https://github.com/vistadataproject/documents/tree/master/Background#technologies

nodeVISTA Stack

Clean, Modular, Separation of Functionality

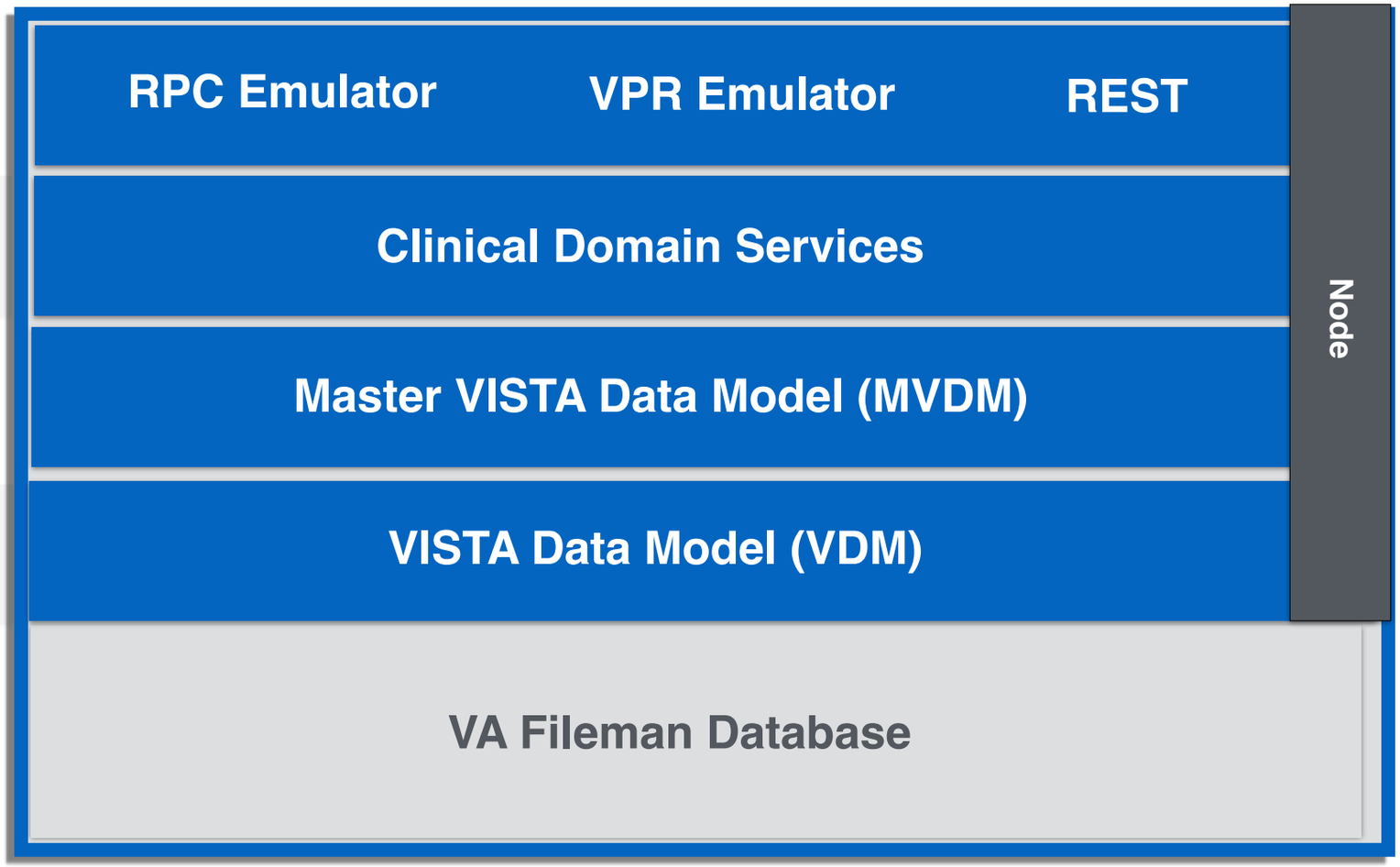
- Emulation and New Interfaces
- All reduce to same service interactions

- (Problem, Pharmacy …) Services over MVDM
- Patient level selection and security

- Normalizes VDM
- Distinguishes Veteran and Patient/Clinical specifics
- A Clean “CRUD+R”/Events paradigm

- Transparent JSON of the native model
- Read 100% data in FileMan
- Write Tested for MVDM covered classes

- All interaction through formal FileMan API
- Only FileMan changes fix Data Dictionary (DD)

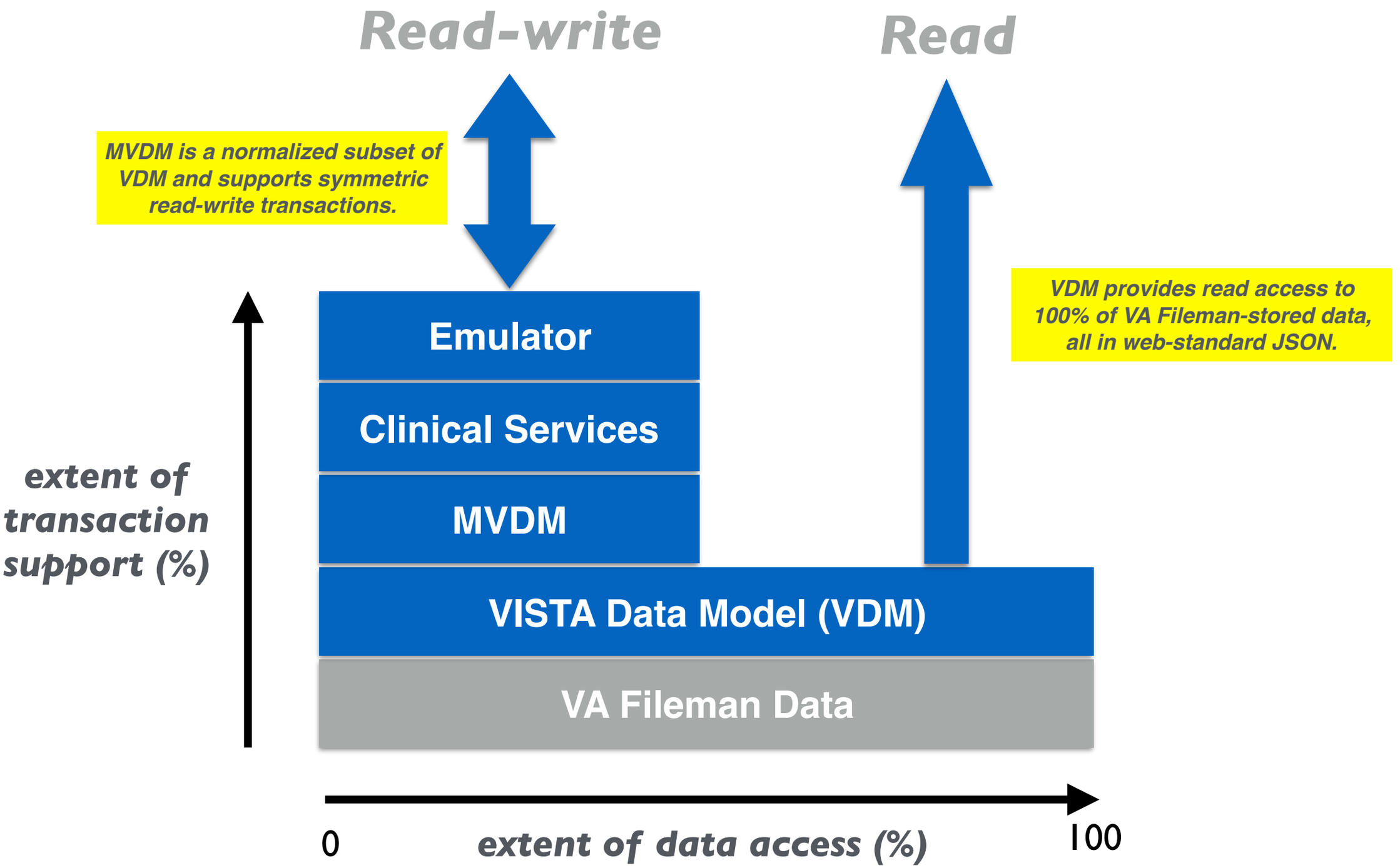


Structured VISTA Server
(mainstream, modular, extensible)

Javascript/Node.js

nodeVISTA Data Access

Starting point is read access to 100% of FileMan-stored data (including labs) and then extends to transactional data



Master VISTA Data Model

Interfacing Evolution

Interface	MUMPS RPCs (x3500)	Master VISTA Data Model (x1)
Method	🚫 Relies on over 3500 client-specific, non-interchangeable legacy MUMPS routines 🚫 Distinct, unique routines for reading vs writing the same data 🚫 Requires extensive knowledge and experience with MUMPS and VISTA	✅ 🆕 Data Model-Driven ✅ 🆕 Client-agnostic ✅ 🆕 One single, symmetric read-write mechanism for all data. ✅ Requires no knowledge or experience with VISTA internals or MUMPS.
Ease of interfacing to new clients	🚫 HARD	✅ EASY
Security	🚫 Patchy, Opaque	✅ Comprehensive, Clear
Authentication	Kernel Access/Verify	Kernel Access/Verify
Access Control	🚫 Dependent on legacy terminal interface Menu Options	✅ 🆕 Applicable to *any* new interface. ✅ 🆕 Data-Centric; ✅ 🆕 Patient-Centric, ✅ 🆕 Attribute-Based Access Control (ABAC)
Fileman API Compliant	🚫 Unreliable, Incomplete 🚫 Variable compliance	✅ Reliable, Complete ✅ 100% Compliant
Audit	🚫 Incomplete 🚫 Bypassess Fileman auditing	✅ Comprehensive AND ✅ 🆕 Patient-Centric
Unit Tested	🚫 NO 🚫 0% logic tested	✅ YES ✅ 100% logic validated
Documentation	🚫 Incomplete, inconsistent, unclear. 🚫 Requires understanding MUMPS code	✅ Complete, consistent, clear. ✅ 🆕 Core is machine generated

Master VISTA Data Model

Features

VISTA Data	Details
Access	**A single, universal, industry-standard mechanism for reading and writing *all VISTA data*.** This mechanism is unified through a read model and write write model integrated into a single, symmetric-read-write data model (VDM), with all data in industry-standard web formats. *This overcomes the well understood shortcoming with VISTA Data Read and Write, which uses completely unique code, models, and mechanisms for reading data as distinct from writing data. Furthermore, the 20+ year old RPCs - over 3300 MUMPS routines which encapsulate all these idiosyncratic approaches (written *exclusively* and in lock-step with the the Delphi code of CPRS, and none of which are documented or maintained) simply cannot be relied on going forward, particularly for generic, external non-CPRS interfaces and clients.*
Integrity	**Comprehensive, automated, standardized, strict data integrity enforcement for *all VISTA data*.** *This is a major improvement over the hodgepodge of legacy, ad-hoc methods that have accumulated over the past 35 years (HL7, RPCs, MUMPS, procedural code), none of which are documented, and all of which are inconsistent, unpredictable, and highly permissive*. See also: Master Data Management
Security	**Comprehensive, industry-standard, fine-grained, data-centric security for *all VISTA data*.** Currently VISTA provides security for only a small fraction of its data, and does this through highly nonstandard, complex, opaque, and unmaintainable methods. Data-centric, attribute-based security is the foundation for all other security levels and technologies, because without knowledge of the data and its attributes, it will not be possible to provide the appropriate security measures on the data. Through metadata enrichment of the VISTA Data Model, VISTA will know *what categories of data it is managing* and thus allow, for the first time, comprehensive, data-centric, attribute-based security "on-the-data" for all VISTA data, permitting the secure exchange of data. See Data-Centric Security, Logical Security, Semantic Security and Attribute-Based Access Control (ABAC)

Master VISTA Data Model

Attributes

Representative	**VDM operationalizes all relevant VA VISTA data to the maximum extent available.** The VISTA Data Model comprises the current existing data-driven architecture of VISTA, and thus leverages all existing VISTA definitions. There is 100% correspondence and coverage of the internal data definitions of any local VISTA and that of its corresponding VISTA Data Model (VDM), since these are maintained always in-sync and up-to-date. Any and all enhancements to any VISTA system and its data definitions will automatically be reflected in the VISTA Data Model through automated, triggered updates whenever VISTA's data dictionary is updated.
Real-Time	**VDM is operationalized using Best-of-Breed real-time server-side runtime technology.** The same runtime technology that runs the largest commericial real-time high-traffic websites such as Walmart, eBay, PayPal, Netflix, Uber, LinkedIn, and the New York Times also runs MVDM. *This maximizes transactional processing performance directly on the transactional database.*
Noninvasive	**VDM provides VISTA with essential new functionality within the current VISTA architecture 'as is', without modification.** No existing VISTA code, routines, packages, modules, infrastructure, or functionality will be affected or changed in any way (i.e. this is a 'safe'and 'noninvasive'). This keeps all existing functionality, while offering new, essential functionality for parallel development of all new web-oriented clients. In addition, it makes it easy and 'safe' to install, as this does not affect any current code or functionality.
Self-Contained	**VDM runs entirely server-side, embedded directly on the existing VISTA database.** This eliminates all moving parts and maximizes transaction processing performance by running as an embedded process directly on the local database, leveraging the 'as-is' database architecture. *This makes it easy to deploy, maintain, and keep highly performant. No moving parts. No external dependencies. No middleware.*

Master VISTA Data Model

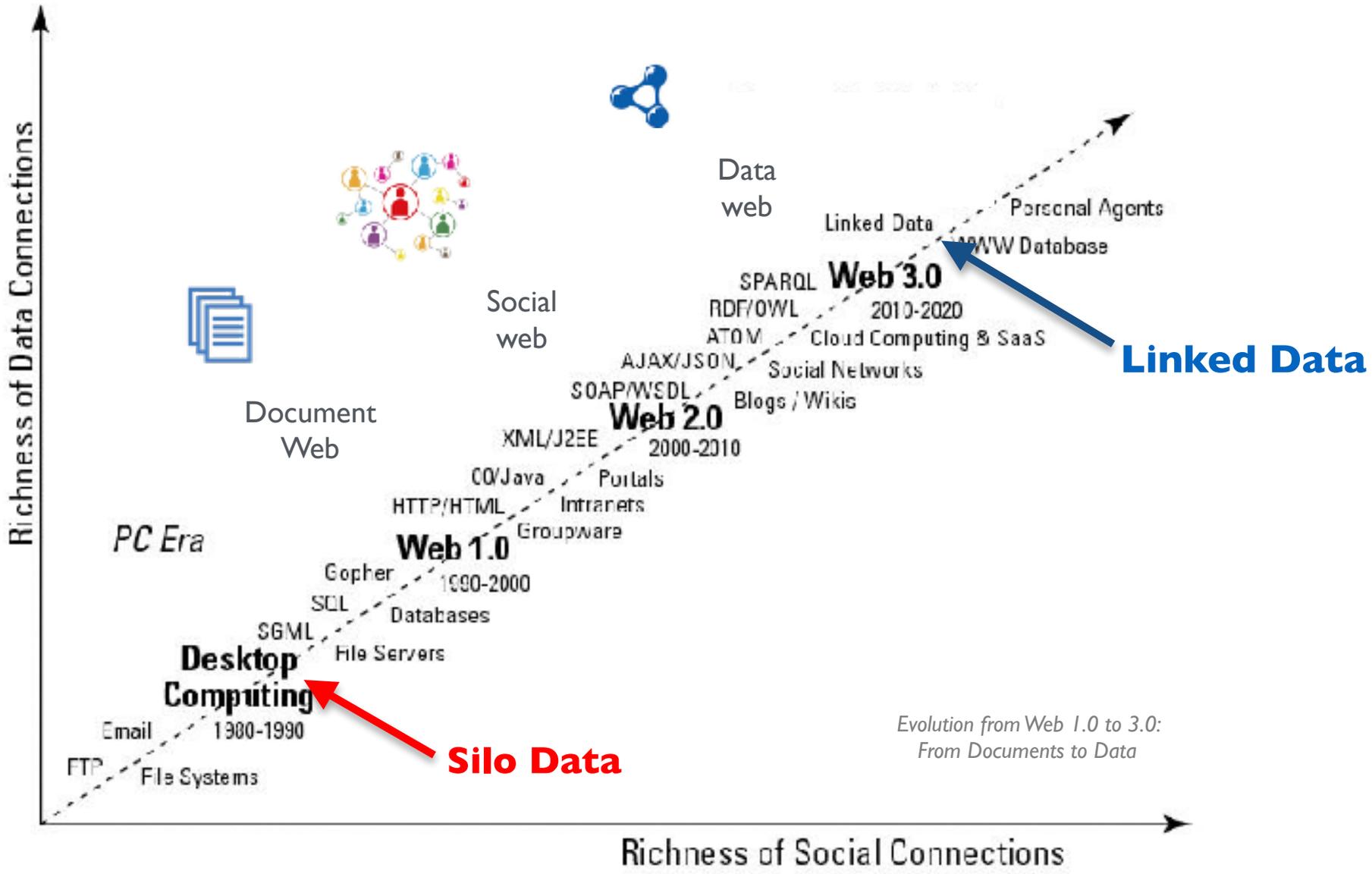
Attributes (continued)

Self-Contained	**VDM runs entirely server-side, embedded directly on the existing VISTA database.** This eliminates all moving parts and maximizes transaction processing performance by running as an embedded process directly on the local database, leveraging the 'as-is' database architecture. *This makes it easy to deploy, maintain, and keep highly performant. No moving parts. No external dependencies. No middleware.*
Data-Centric	**VDM is a completely new, purely data-centric approach to managing VISTA's data**. It does not involve changing a single line of VISTA's existing M procedural code, nor is it 'wrapping' (i.e. secretly using) any legacy code, routines, or RPCs dressed up within a shiny new programming language or encapsulation mechanisms, which add yet more layers of obfuscation on the data. A data-centric approach ***comprehensively exposes all the data, which exposes the fact that VISTA has a data model*** - which up to this point has not been realized nor taken advantage of. *This is the opposite of a code-centric approach, which obfuscates the data and its data model*.
Web-Standard	**VDM technologies are 100% web standard** and all used in production settings by the worlds' largest corporations and organizations. For further information see standards and technologies.
Empiric Evolution	**VDM employs a new approach to emprically evolving VISTA's capabilities through rapid, iterative, functional prototypes.** This allows the focus to remain on exploration of new techniques and approaches, rather than on more superficial end-user requirements, which rarely if ever attempt to tackle the deep conceptual and technological issues of data management. This is *the opposite waterfall development*. See spiral model

Linked Data

- *Evolution of Data: Increased connectivity*
- *What is it Linked Data?*
- *What problems does it solve?*
- *Who uses Linked Data?*
- *Health Data: Many diverse models*
- *Linked Data: Accommodates model diversity*
- *Health Data: PCAST Recommendation*

Evolution of Data: Increased Connectivity



Evolution from Web 1.0 to 3.0: From Documents to Data

VISTA began as the Decentralized Hospital Care Program (DHCP) in 1981, before the existence of the Internet. After the Internet protocol emerged in 1988, the World Wide Web rapidly evolved from a document web to social web to data web. To bring any system or data into today's world wide data web, it needs to be web data standards compliant. (Linked Data).

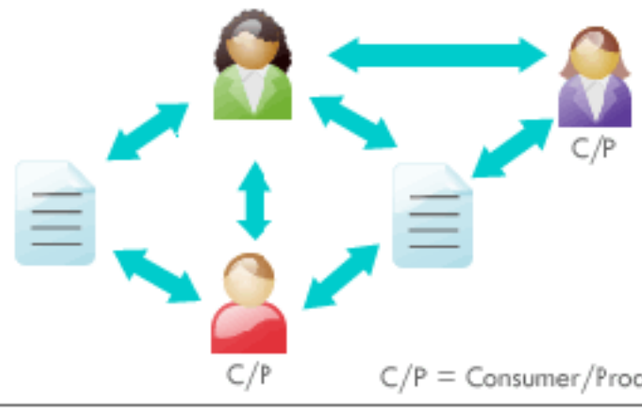
http://en.wikipedia.org/wiki/History_of_the_Internet
http://paanchiweb.blogspot.com/2012/12/getting-essence-of-semantic-web.html

Evolution of the Web: ***From Documents to Data***

Web 1.0

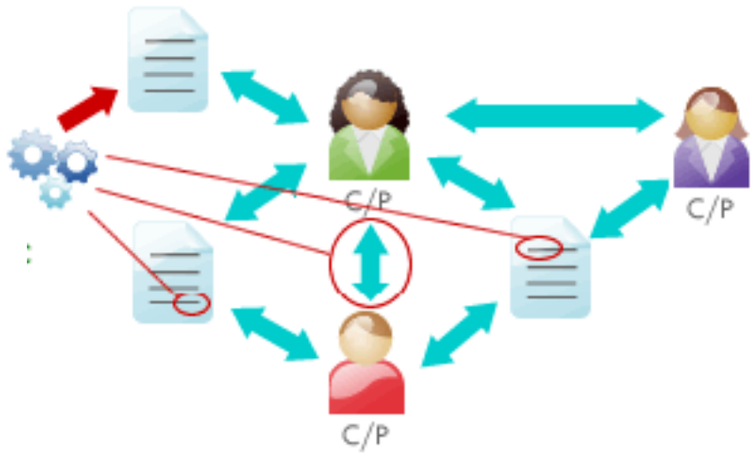
Linked Documents
Document Web (HTML)
Read-only web (**humans** only)

Web 2.0



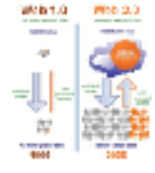
Linked People
Social Web
Read-write web (**humans** only)

Web 3.0



Linked Data
Semantic Web (RDF)
Read-write web (**machine processable**)

http://tomgruber.org/writing/CollectiveKnowledgeSystems.htm



Linked Data: ***What is it?***

The World Wide Web (W3C) Standard for semantic information integration

HTML (hypertext markup language)
For **humans** to exchange information

enables →

Linked Documents
(Document Web)

RDF (resource description framework)
For **computers** to exchange information

enables →

Linked Data
(Semantic Web)

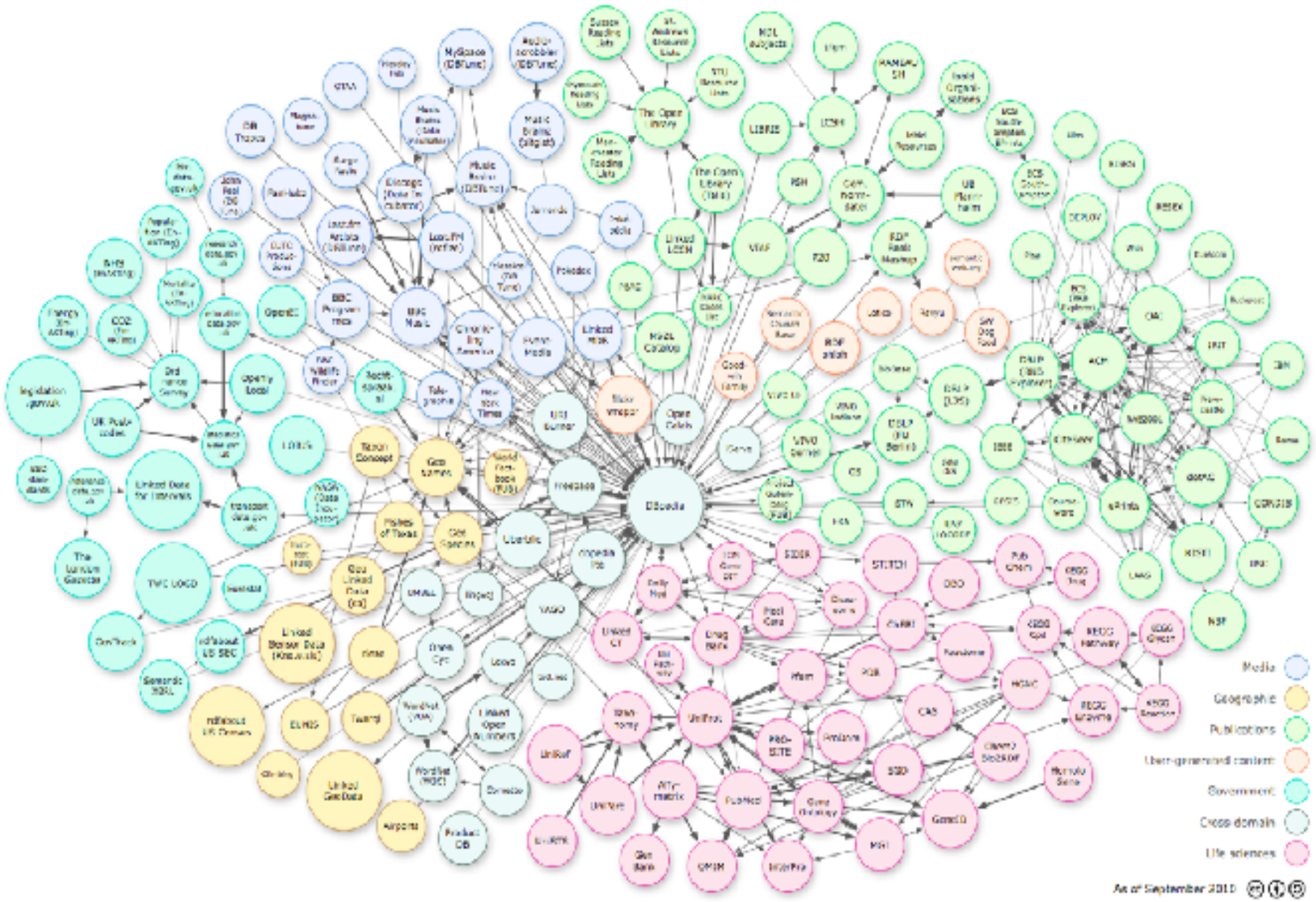
"The Semantic Web [Linked Data] provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries."

Tim Berners-Lee, MIT Professor and Inventor of the World Wide Web

As a W3C standard this supports Internet-scale data integration.

Linked Data: ***What does it enable?***

Web-scale semantic integration of data



Linked Data

This figure shows the Linked Open Data (LOD) cloud, which semantically links hundreds of Linked Data sources including Media, Geographic, Government, and Life Sciences databases.

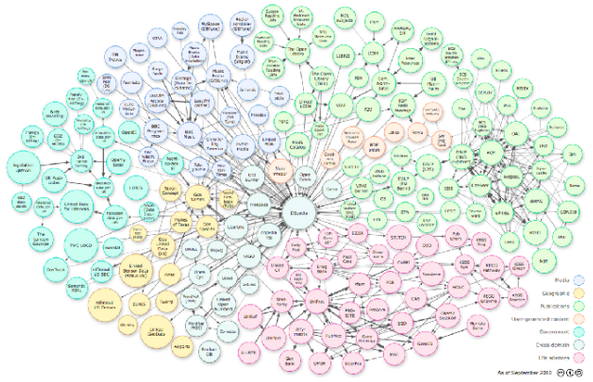
Each circle represents one data source or database. These are semantically linked to other data sources, creating a single virtual federated internet-scale database.

At the center of is DBPedia, the Linked Data version of Wikipedia, which is semantically linked to hundreds of data sources.

Linked Data: ***Who Uses It?***

The Linked Data approach to ***Internet-scale semantic data integration and search*** *by the world's largest data management organizations such as Google, LinkedIn, Facebook, and IBM Watson.*

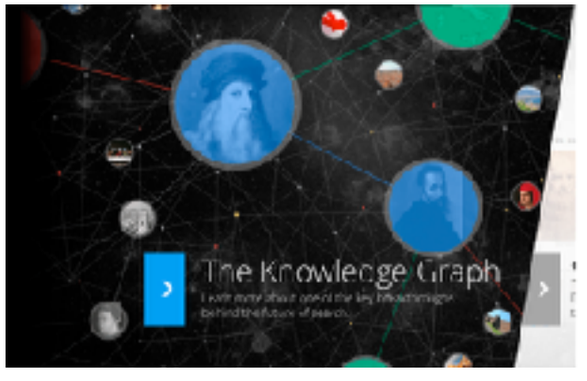
Wikipedia
(knowledge graph)



Watson
(knowledge graph)

LinkedIn
(professional graph)

Facebook
(social graph)



Google
(knowledge graph)

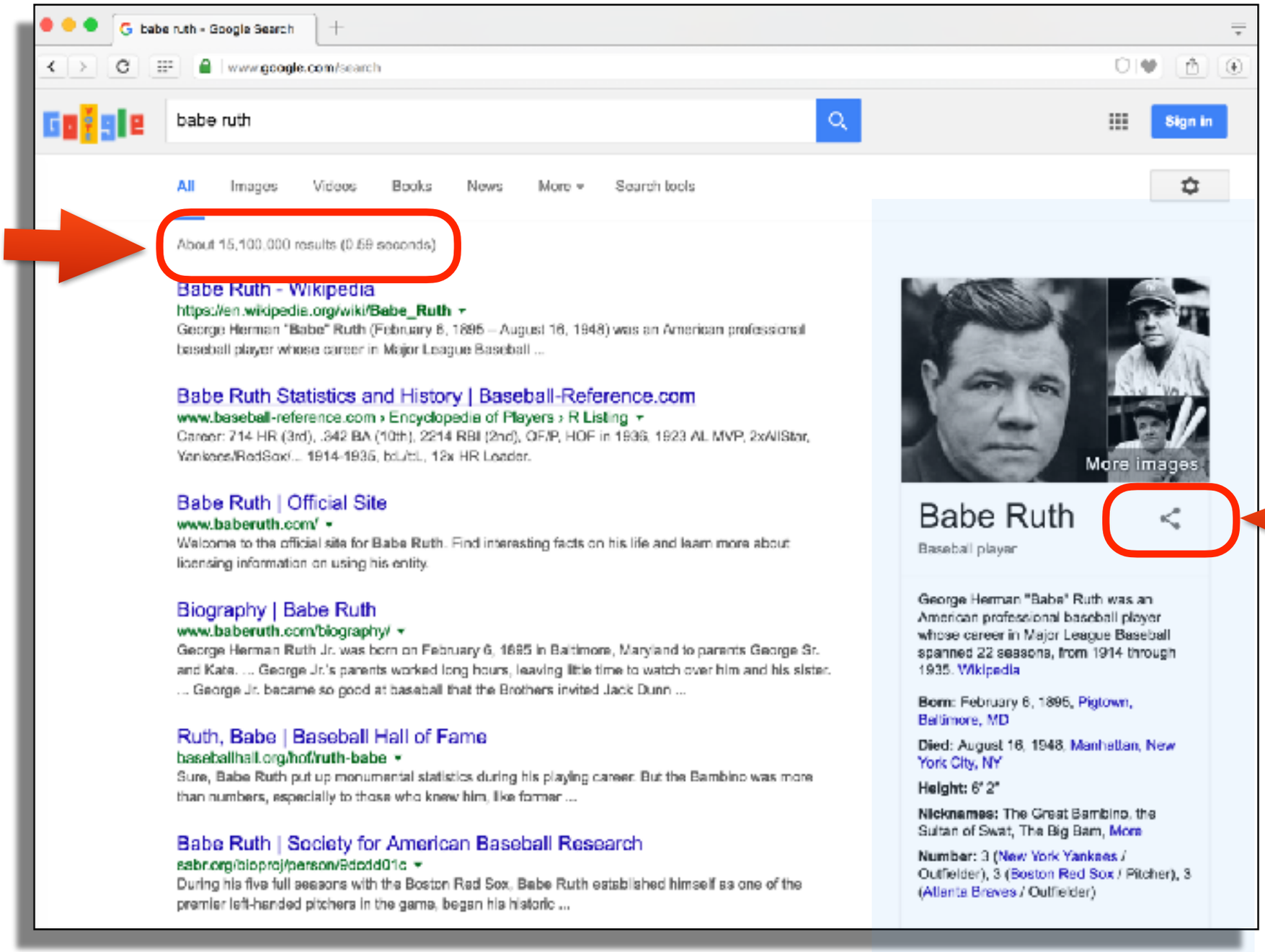
Linked Data: Semantic Search

All the major search engines (Google, Yahoo, Microsoft, Yandex…) use the same shared web schema in RDF to index, search, and structure all data on the web, making it all semantically searchable.

From Strings to Things

Google Guess

Over 15 million guesses. String matching. Statistical page rank algorithms. "Black box"



Google Know

Single, exact, semantic result. Based on on world wide knowledge graph of Linked Data. (See Linked Data symbol)

Strings **Things**

U.S. Healthcare: PCAST Recommendations

REPORT TO THE PRESIDENT
REALIZING THE FULL POTENTIAL OF
HEALTH INFORMATION TECHNOLOGY
TO IMPROVE HEALTHCARE
FOR AMERICANS:
THE PATH FORWARD

Executive Office of the President
President's Council of Advisors
on Science and Technology

"The best way to manage and store data for advanced data analytical techniques is to break data down into the smallest individual pieces that make sense to exchange or aggregate. These individual pieces are called "tagged data elements," because each unit of data is accompanied by a mandatory "meta data tag" that describes the attributes, provenance, and required security protections of the data.

The indexing and retrieval of metadata tagged data, across large numbers of geographically diverse locations, is an established, highly developed, technology—the basis of web search engines, for example".

- ***Linked Data (RDF) is the World Wide Web standard for semantic metadata tagging for data on the web, used by all major search engines.***

http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-health-it-report.pdf

U.S. Healthcare: PCAST Recommendations

UNITED STATES
DEPARTMENT OF VETERANS AFFAIRS

Search All VA Web Pages | Search | » Open Advanced Search

Home | Veteran Services | Business | About VA | Media Room | Locations | Contact Us

- One-VA TRM Home
- Search One-VA TRM
- Technology/Standard List
- TRM Reports
- Submit Idea to Improve TRM
- TRM Glossary
- Site Map

ONE-VA TECHNICAL REFERENCE MODEL V15.1

TRM **Resource Description Framework (RDF)** ← ***(Linked Data)***

General | Decision | Reference | Component | Category | Analysis

General Information

Technologies must be operated and maintained in accordance with Federal and Department security and privacy policies and guidelines. More information on the proper use of the TRM can be found on the TRM Proper Use Tab/Section.

Website: Go to site

Description: The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications. They were originally designed as a metadata data model. It has come to be used as a general method for the conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats.

Decision: View Decisions

Decision Constraints:

Decision Source: TRM Mgmt Group

Decision Process: One-VA TRM v13.10

Decision Date: 10/25/2013

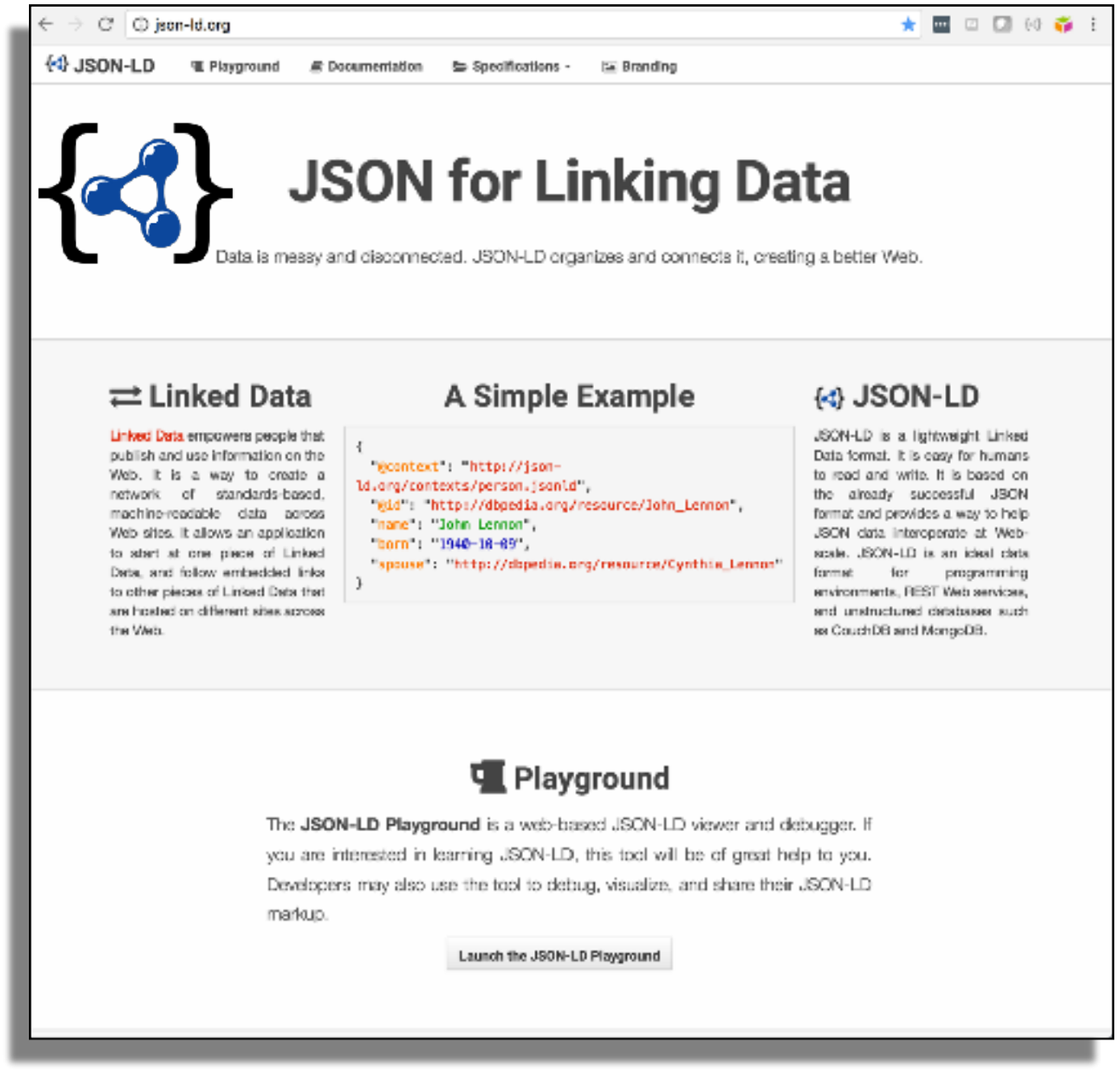
Introduced By: TRM Request

Standards Body: W3C

http://www.va.gov/TRM/StandardPage.asp?tid=6405#

Linked Data: JSON-LD

The Resource Description Framework (RDF) has many serializations. The form most commonly used for web applications is JSON-LD,



Resources

Linked Data: ***Accommodates both Standards and Innovation***

Information Models:
An apparent conflict between standardization and innovation

Standards: need to remain static in order not to be disruptive for adopters.

- Static, Brittle
- Centralized
- General (Common Denominator)
- Committee-driven
- Large, "all-or-nothing", disruptive updates

VS.

Innovation: requires continuous evolution of thousands of new information models.

- Adaptive / Evolutionary
- Decentralized
- Highly Specialized, "Best of breed"
- End-user / specialist – driven
- Small, continuous, low-impact updates

What are the options?

Centralized, Model-rigid approach: For exchange of information to occur all models must remain fixed, and data must go through only one central 'broker' model..

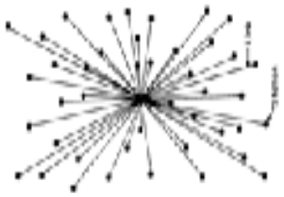
Our current approach to healthcare data

Decentralized, Model-flexible approach: Multiple models peacefully co-exist and evolve, mediated by their ability to freely link to any model at all times. In this approach, all models are free to evolve AND are capable of resolving to a common standard model at all times. *The only current technology that supports this granular data-oriented approach is RDF (Linked Data).*

Linked Data supports <u>both</u> standardization and innovation

http://www.carlsterner.com/research/2009_resilience_and_decentralization.shtml

Data Integration: ***Legacy vs. Linked Approach***



Architectural Issues	**Current Approach**	**Linked Data (RDF)**
Data model characteristics	**Model-rigid.** Only one lowest common denominator model unifies information. Must remain unchanged to orchestrate. **Restrictive expression.**	**Model-flexible.** All data models may independently evolve. **Maximizes expressivity.**
Data model compatibility	**No model diversity permitted** A one-size-fits-all mega-model	**Multiple models peacefully coexist** Data model flexible
Data model evolution	**Costly and difficult to evolve models.** Due to model-rigid architecture.	**Cheaper and easier to evolve models.** Due to model-flexible architecture.
Data access method	Downloading + Aggregating	**Linking + Federating**
Scalability: incremental effort required to add new data sources	**Common model** must be updated	**Individual models** may be independently and incrementally semantically linked.
Primary Function	**Data Syntax and Transport**	**Data Model**
Granularity	**Document-centric** (message-centric)	**Data-centric** (individual data elements)
Semantics	**Weak semantics.** **Extrinsic, separate from the data.** Depends on an external data model.	**Strong semantics.** **Intrinsic, integrated with the data.**

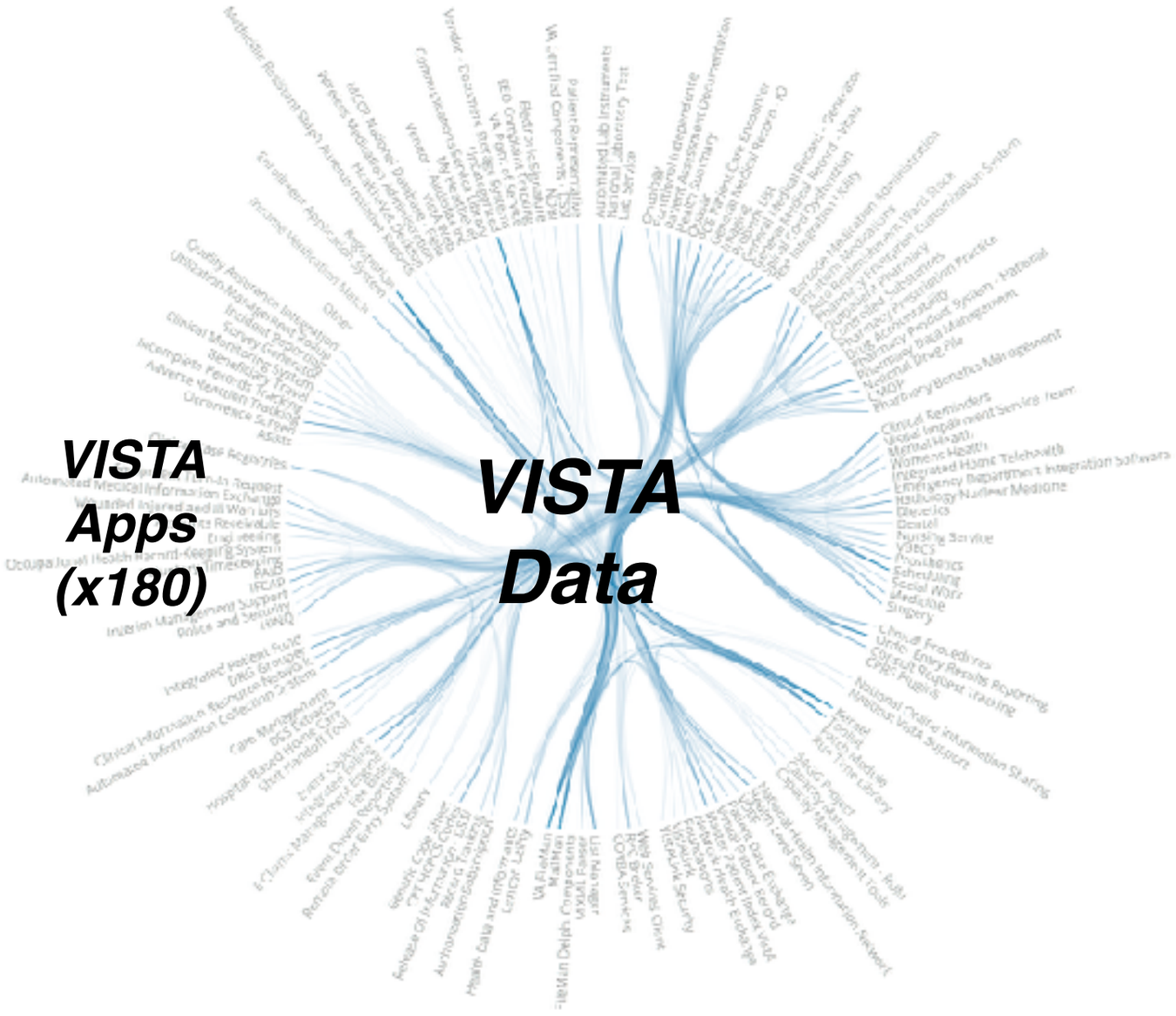
Links: ***Linked Data***

W3C Linked Data Standard
http://www.w3.org/standards/semanticweb/data
W3C Linked Data Platform
http://www.w3.org/TR/ldp/
W3C Semantic Web Healthcare and Life sciences
http://www.w3.org/blog/hcls/
HL7-RDF Healthcare Standards Work Group
http://wiki.hl7.org/index.php?title=RDF_for_Semantic_Interoperability
Semantic Web
http://en.wikipedia.org/wiki/Semantic_Web
Linked Data: Tools (>1000)
http://www.mkbergman.com/sweet-tools
Linked Data: Adaptors (100's)
http://www.w3.org/wiki/ConverterToRdf
Linked Data: Roadmap for Healthcare Interoperability
http://yosemiteproject.org

VISTA Internal Integration

Fully integrated by design.

All Apps.
All Data.
Integrated.
Real-time.



Comprehensive Patient-Centric Integrated EHR

The data architecture of VISTA consists of 180+ modules for clinical care and administrative functions integrated within a single, shared, integrated database.

VISTA applications (outer ring) all share the same, single, authoritative data (inner circle). All data between applications is integrated within this single multidimensional database (connecting lines).

Caveat: Data remains integrated only so long as it remains within to the VISTA/ M environment.

VISTA Interfacing

Code-centric

Opaque.
Brittle.
Inconsistent.
Insecure.

Code-centric interfacing manages VISTA as a "Black Box"

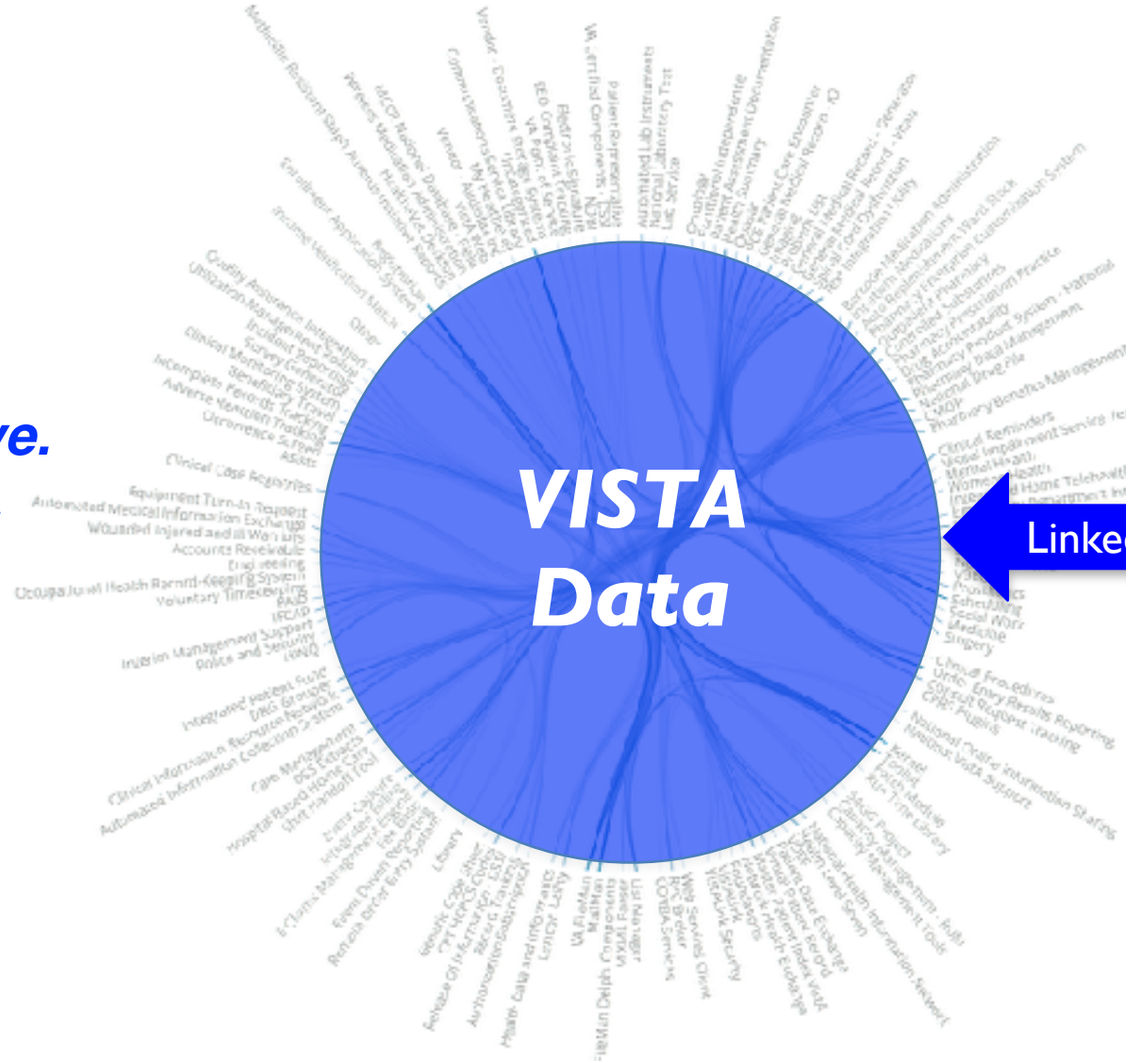
Code-centric interfacing has no logical connection to the internal structures, context, or definition of the data within VISTA. Rather, the code obfuscates the native data model and structures by encapsulation.

As a result, code-centric interfacing lacks any uniform method to comprehensively or securely interface to VISTA data. There are infinite permutations of hard-coded interfaces possible.

VISTA Interfacing

Model-centric

Comprehensive.
Transparent.
Standard.
Secure.



Linked Data Model

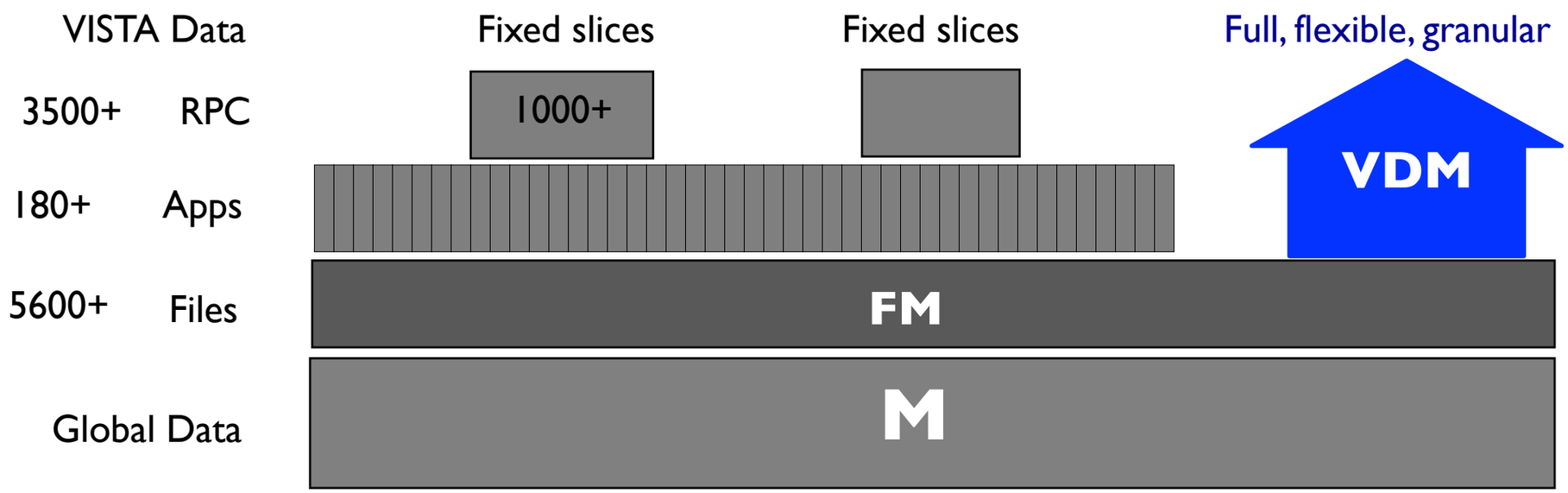
Model-centric interfacing makes all VISTA data accessible.

The web-standard Linked Data model logically connects to all data and structures ***internal*** to VISTA, allowing secure read-write ***external*** to VISTA with one web-standard mechanism.

Representing VISTA's internal data model as Linked Data Model model enables web-scale external interfacing and integration.

VISTA Interfacing Evolution

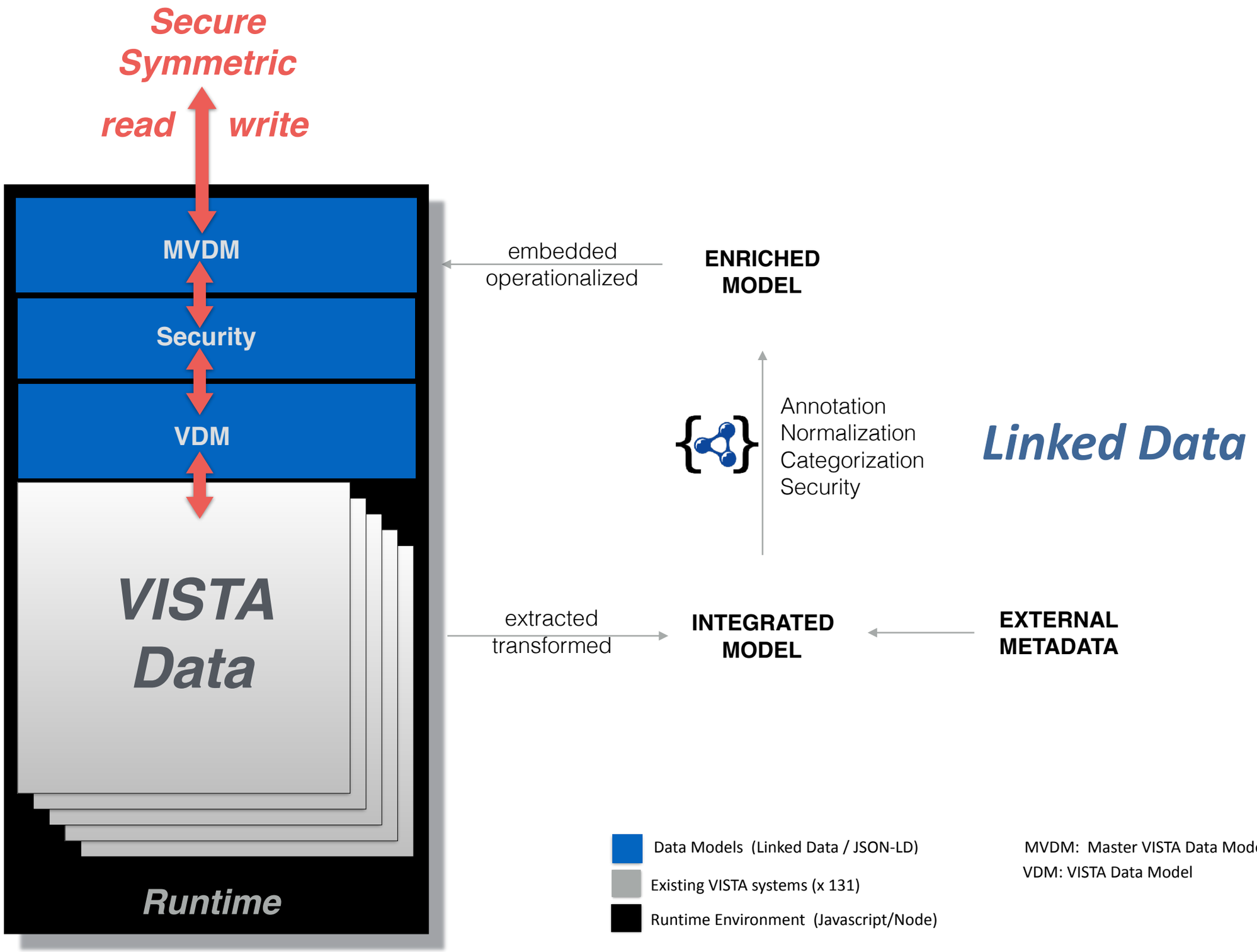
	CPRS	JLV	VDM
View	Delphi	HTML	HTML
Controller	Delphi	Java	Javascript
Model	Undefined	Undefined	**Full VISTA Model**
Read/Write	Asymmetric	Read-only	Symmetric
Format	proprietary	proprietary	JSON-LD
Method	MUMPS code	MUMPS code	Query



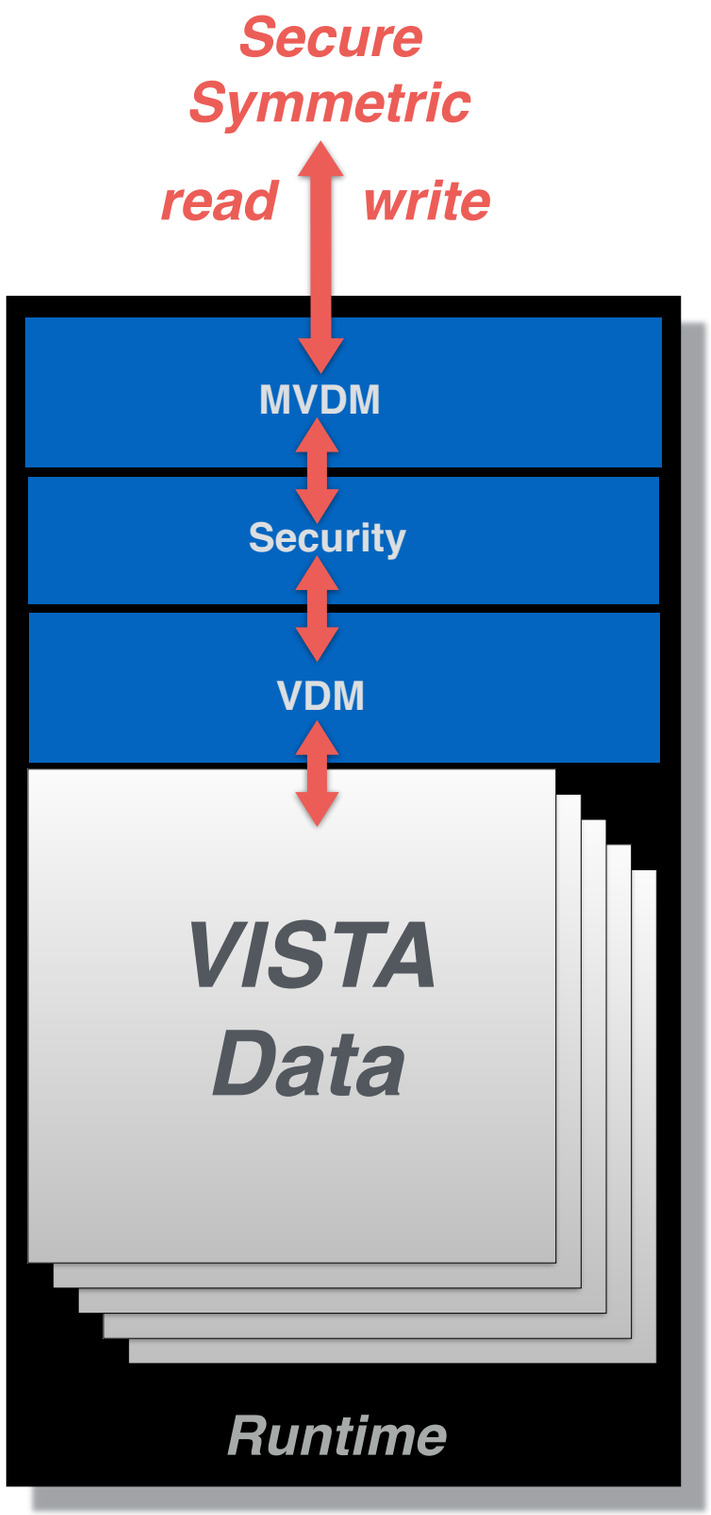
VISTA Data Model (VDM) can access all data spanning 180+ applications with full granularity and definition because the fully exposed VISTA Data model logically bridges all applications through their native data dictionaries. No legacy APIs, HL7, RPCs, or MUMPS code. Just data. All of it. Defined. Structured. Secure.

Resources

VISTA Data Model Development



VISTA Data Model Detail



Linked Data Model
- Industry-standard machine-processable web data model
- Uses schema-backed JSON with Linked Data extensions (JSON-LD)
- All VISTA data models are expressed, processed, and enriched as JSON-LD.

Master VISTA Data Model (MVDM) (x1)
- A subset of VDM that is normalized across all VDMs
- Incorporates all functionality of the Security Model
- Incorporates all functionality of the VDM
- Supports remote secure read-write across all VISTA instances
- Supports Master Data Management across all VISTA instances for any specified data category

Security Model (x1)
- Provides data-centric logical security model for all VA VISTA data.
- Provides data-centric security based on data attributes and categories
- Specifically provides “on-the-data” granular patient-centric data security.

VISTA Data Model (VDM) (x131)
- Represents the full native operational data model of any local VISTA
- Enables comprehensive access to all VISTA data (all 65,000+ data fields)
- Is enriched by additional metadata and logic to support write back
- Provides native symmetric read-write to any local VISTA
- Eliminates need to know anything about VISTA code or internals

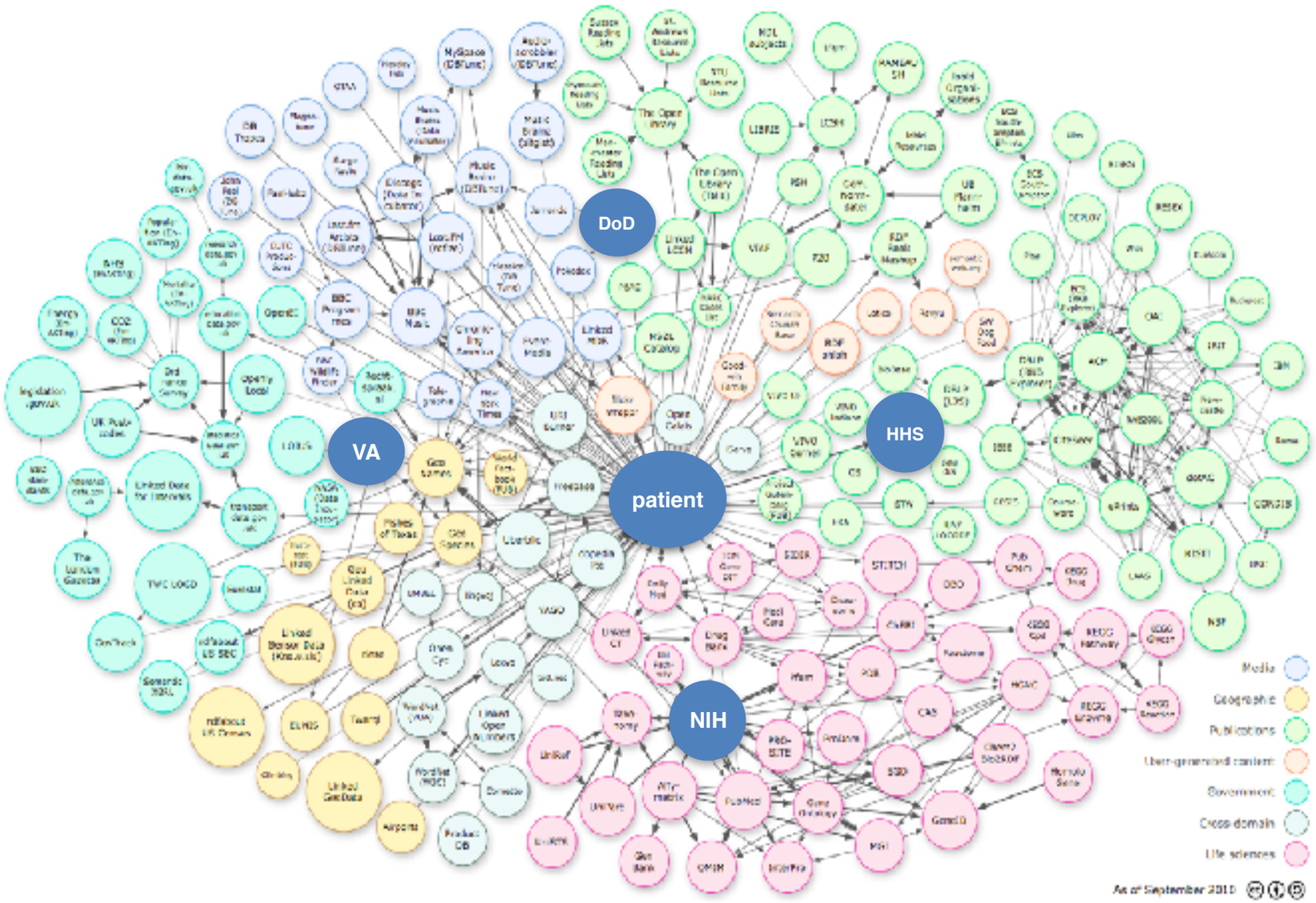
VISTA Systems (x131)
- Each contains over 35 years of VA clinical and institutional data

Runtime Environment (Javascript / Node)
- Industry-standard Node.js server-side runtime environment
- All data models and data transformation run in-process, server-side
- All read-write transactions run in-process, server side

Data Models (Linked Data / JSON-LD)

Application: Web-scale data integration

Web-scale semantic integration of public, private, clinical, and health sciences data.



Linked Data Integration

This figure shows the Linked Open Data cloud, which semantically links billions of facts from thousands of data sources including Media, Geographic, Government, and Life Sciences databases.

Clinical and health data such as VA, DoD, and private sector patient data may also be natively integrated with this data cloud in Linked Data form.

Appendix II

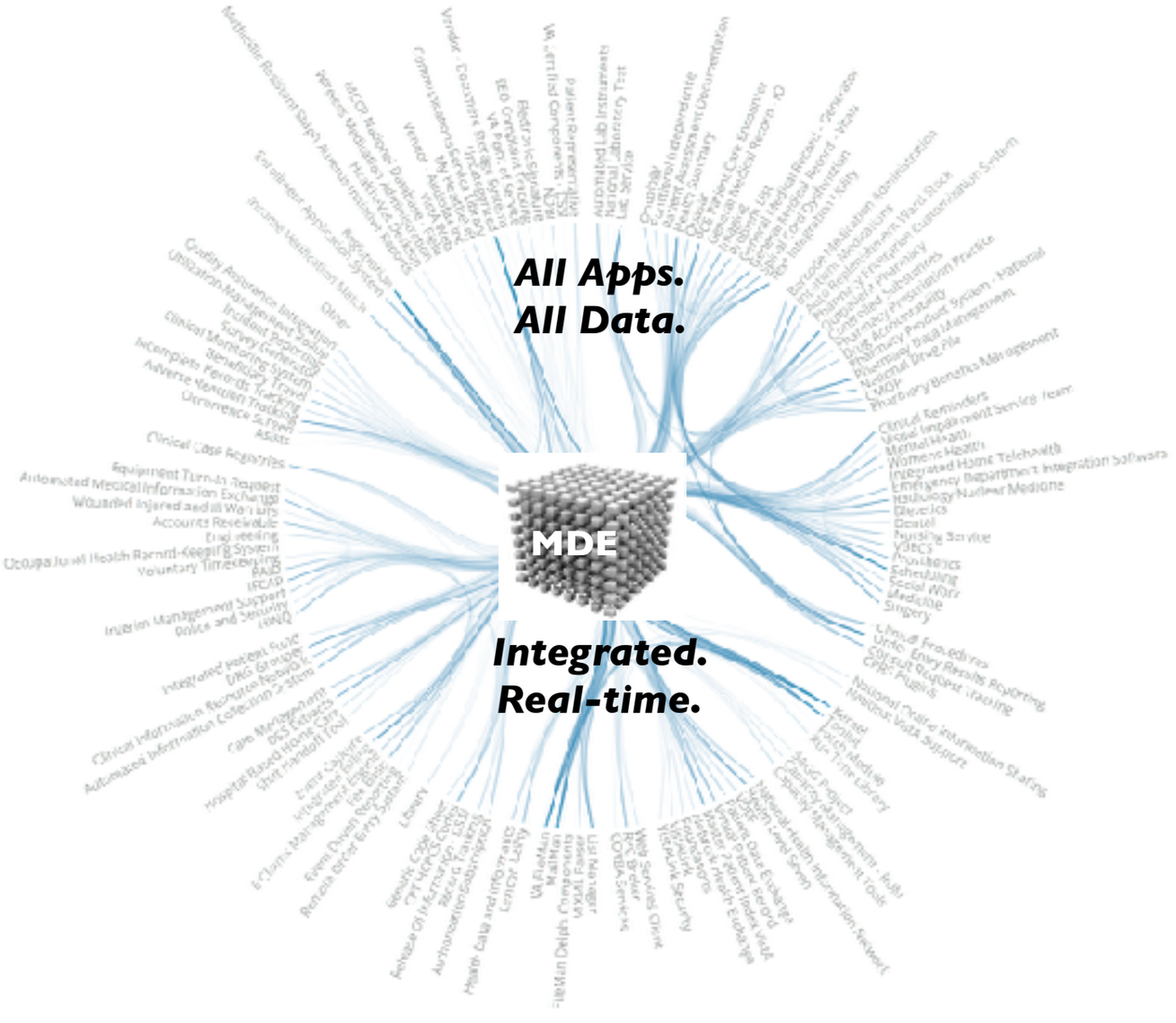
VISTA Data Model (VDM)

A Path to VISTA Data Management:

- VISTA's Database
- VISTA's Data Model
- VISTA's Data Model Exposed
- Benefits of Leveraging VISTA's Data Model:
  - Master Metadata Management
  - Centralized Knowledge Management
  - Master Data Definition
  - Patient-Centric Security Model
  - Separating Business logic from Data
  - Query Access

Review: VISTA's Database

The foundation of VISTA is a high performance Multidimensional Data-integrated Application Engine in which all data and all applications are fully integrated in real-time with each other and to one single authoritative data source.



VISTA's integrated application data engine. *All 180+ applications are integrated with their data and logic inside the multidimensional data engine (MDE). This keeps transactional patient data and logic highly integrated for real-time use within one single data store.*

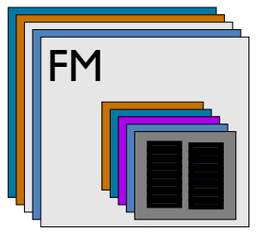
VA uses the same healthcare industry-standard data engine that nearly 50% of hospitals in the US currently use as their core EHR database.

This same multidimensional data engine is also used as the real-time transaction engine of the world's five largest banks, the largest Wall Street trading systems, and over half a million ATMs in North America.

Review: VISTA's Data Model (VDM)

All real-time transactional operations in VISTA take place within the the multidimensional data engine (MDE). To provide consistent structure and model for the data, a data dictionary driven hierarchical data storage system is overlaid on top of the MDE. All VISTA applications read and write data to this hierarchical store using a file manager called Fileman.

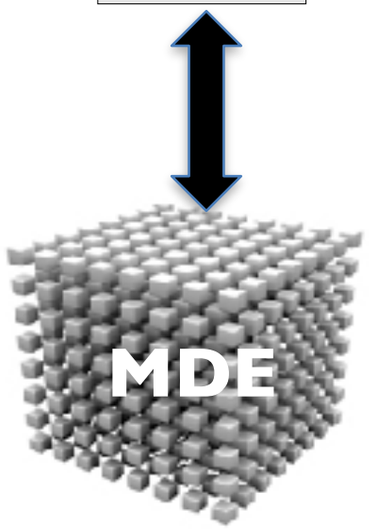
File Manager



Hierarchical Graph Store

This layer provides the data structures, data models, and data management for VISTA. Fileman is the DBMS for this hierarchical store.

Apps Data

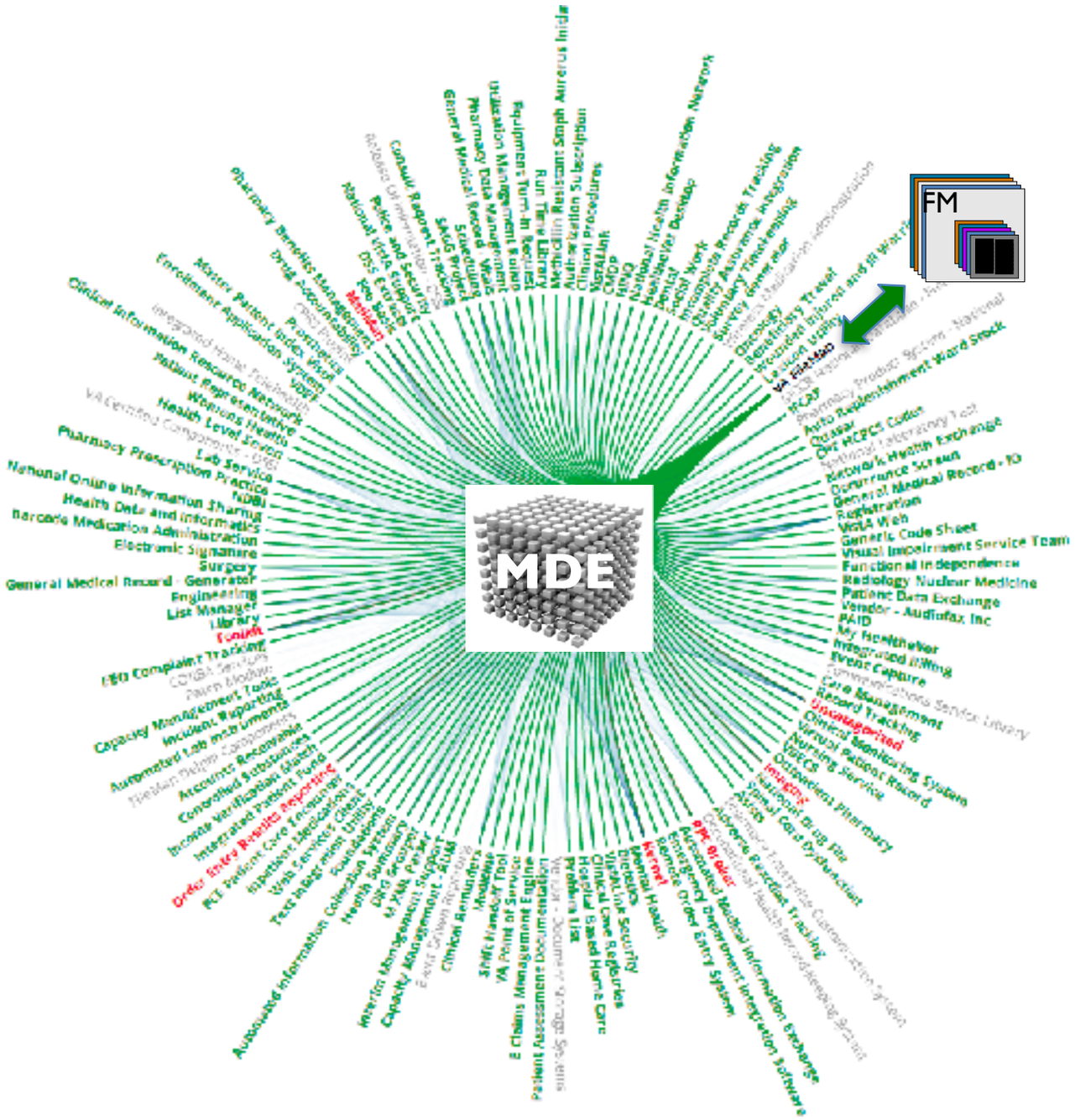


Multidimensional Data Engine

This is the data-structure flexible transactional core of VISTA. This core is a healthcare industry-standard for nearly 50% of the hospitals in the US. This is also used as the real-time transactions engine of the world's five largest banks, the largest Wall Street trading systems, and over half a million ATMs in North America.

Review: VISTA's Data Model (VDM)

All VISTA applications read and write data to VISTA's data engine through the NoSQL hierarchical data manager, Fileman (FM). Unlike many NoSQL databases, VISTA's database is not schema-less, but schema-driven. VISTA's schema is self-documenting through data dictionaries in Fileman. *Surprisingly, this fact has not yet been leveraged to expose and modernize the data model.*



All VISTA data is organized through its data model, accessed through its DD.

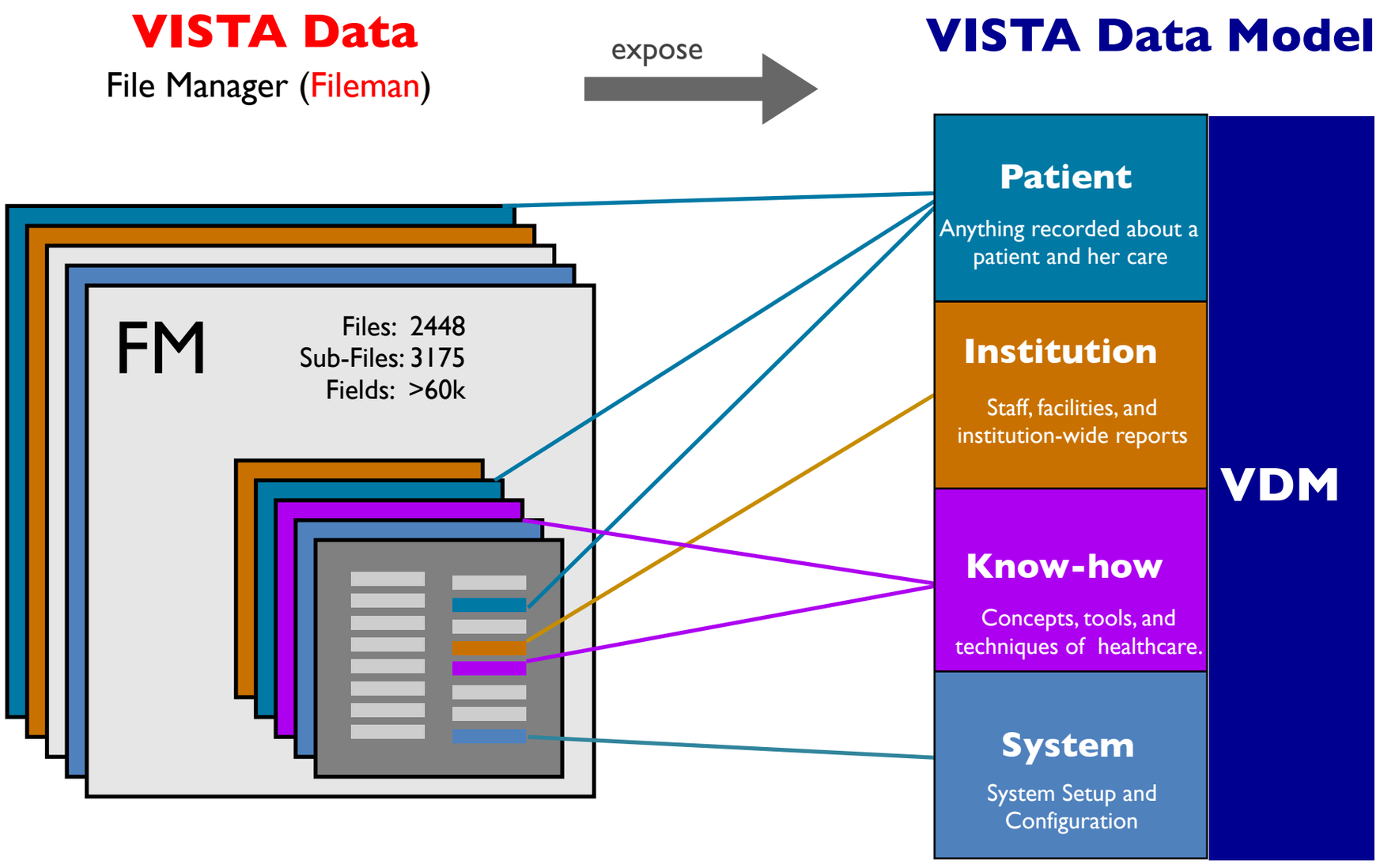
All VistA applications use Fileman to access, structure, and query data in VistA.

The green lines represent the data flow of all VISTA application to Fileman for all read and write operations to the Multidimensional Data Engine (MDE).

Reference: code.osehra.org/vivian

VDM: Expose and Leverage the Model

The first stage of data modernization is to expose and leverage VISTA's real, live operational data model. Since this is just metadata, there is no patient data involved. Unlike many NoSQL databases which are schema-less, VISTA's NoSQL database model is self-documenting through Fileman. This allows us to render this in a standard definition format. In this new web-standard medium, data can be sorted, tagged, searched, and organized by data categories such as by patient, institution, know-how, or system information.



Access: Thousands of RPCs, API's, HL7
Model: Unknown

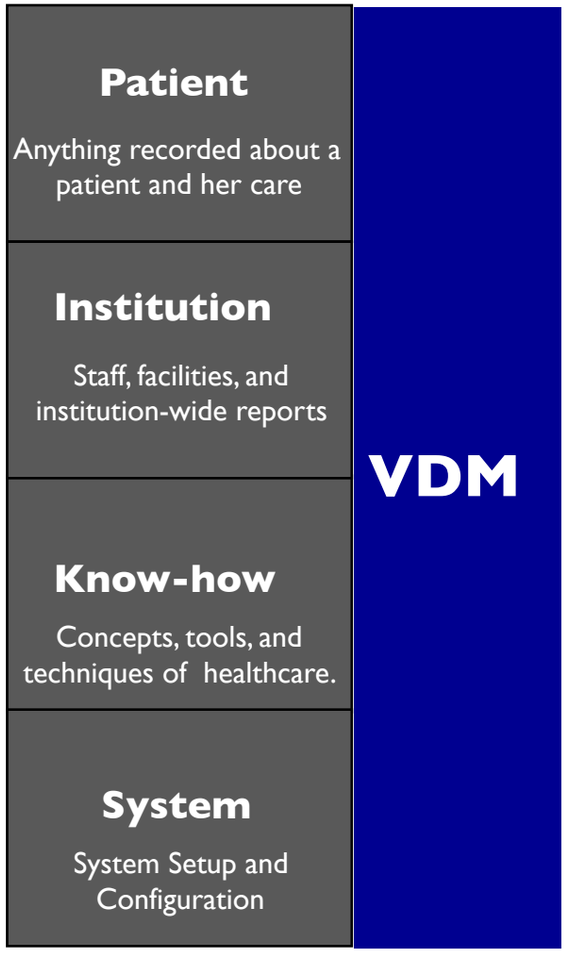
Access: Single query access
Model: Consistent, Transparent

VDM: Master Data Management

A benefit of a VISTA data model allows one to manage data logically across all VISTA application boundaries independent of the source of the data. This lets one manage data with much flexibility, including logically partitioning and managing the data using metadata tagged categories (such as Patient, Institution, Know-how, and Systems information).

- ***One may logically partition data by any class of data***, such as Patient data, Institution data, re-usable Know-how, and System configuration data.
- ***One may extract and move <u>all</u> patient data from system to system with one operation***, making system configuration migration and patient record movement far more efficient.
- ***One may apply security metadata or protocols to any of these logical classes*** of data. For example a patient-centric security model for patient data, and thus enforce patient-centric controls on information exchanged.

VISTA Data Model



Benefits:

Patient Data Management
Extract and manage patient data with patient-specific security and metadata, allowing patient-centric controls on data access and exchange.

Institutional Data Management:
Institution specific data can be exchangeable and centrally manageable

Knowledge Management:
Common medical concepts, standards, and know-how may be identified and managed as a clearly defined class of VISTA data.

System Management:
The entire configuration of a system can be identified, extracted, and transported, and inserted from system to system

VDM: Patient-Centric Data Security

An exposed VISTA data model allows one to tag and partition certain classes of data separately from all other data in VistA. Specifically, this allows one to granularly partition any and all Patient data from all other kinds of data in VistA. This provides true, direct, "on the metal" security on patient data itself.

- *The most important class of data to apply security is the Patient data category. This will allow very granular patient-centric security on the data itself.*

- *Current VISTA security is based on actions one can take using a legacy menu system - which has no relationship to the data.*

- *With a data-centric security model, we can specify not just what type of data ("Mental Health Record"…) but whose data ("For patient X").*

- *This is much more specific and secure than the prior VistA security model.*

Patient-centric security model
Extract and manage patient data with patient-specific security and metadata, allowing patient-centric controls on data access and exchange.

This is compatible with security notions in Meaningful use Data Exchange: it can suppress even data that exists if there is no access permission.

It is essential to improve precision in data security to permit access to VISTA data securely. Otherwise one will have to reverse-engineer 3300 legacy RPCs and their one-off use of Kernel's menu options for each payload.

A patient-centric security model is much more appropriate, flexible, and secure as a foundation for patient data security than the current VistA security model. The current VistA security model provides security only indirectly, through legacy controls of a menu system for a legacy roll and scroll terminal interface – which has nothing to do with the type of data at all (!).

VDM: *Analytics Driving Interoperability*

Comprehensive exposure and analytics of the VISTA Data Model will drive enhanced data use and interoperability as well as a major improvement in the structure of the database itself. To address these and other areas, focused reports could be generated from the model including:

Report	*Activity*	
Inconsistencies between VistA data models	Drive dictionary and code fixes in various centers so that every center is running the same consistent model	→ **Enterprise Data Model**
Isolate centrally and locally managed "know-how"	Enables the next generation of enterprise knowledge services that seamlessly synchronize VistAs and other applications	→ Centralized Knowledge Management
Under-definition in the model	Too many ill-defined string values and not enough nuance ("zip code", "telephone number") can be defined, and provided additional metadata ("home", "work", "mobile") leading to a plan for incremental dictionary improvement	→ **Enterprise Data Definition**
Key logic performed within FileMan	The barrier between the data store and business logic will be laid bare. This report will encourage the movement of certain types of logic into FileMan and out of less maintainable procedural code.	→ Clean separation of business logic from data
Overlooked but highly valuable types of patient data (Some of this may have been hard-set by the application logic, and overlooked by Fileman or DD)	Improve VistA Data mining (for CDW etc.) and interoperability (more comprehensive electronic patient records). Without a complete model, how do you know what you're leaving behind?	→ Clinical Research → **Interoperability**
VistA model/ FHIR comparison (key types)	Show how a direct from VistA transformation can remove the need for redundant intermediate, hard to maintain procedural code	→ Accelerate Data Exchange
Isolate patient from other types of data	Enable patient-data access control rather than the crude option/API security now in VistA	→ **Patient-centric Security Model**

Appendix III

Linked VISTA (Master VDM)

Features

- *Enterprise Cross-linked VISTA Model*
- *Comprehensive Data Access*
- *Web-standard representation language*
- *Web-scale semantic integration*
- *Knowledge Discovery*
- *Native terminology integration*

Linked VISTA: Enterprise Cross-Linked Master VISTA Data Model (MVDM)

Exposing VistA's Data Model (VDM) and applying analytics allows one to cross-link all local VDMs to an enterprise VISTA data model (Linked VDM), providing the capability for Enterprise query and Enterprise data management. This leverages the capability of Linked Data to create a cross-VISTA (enterprise) data model.

Silo VISTAs

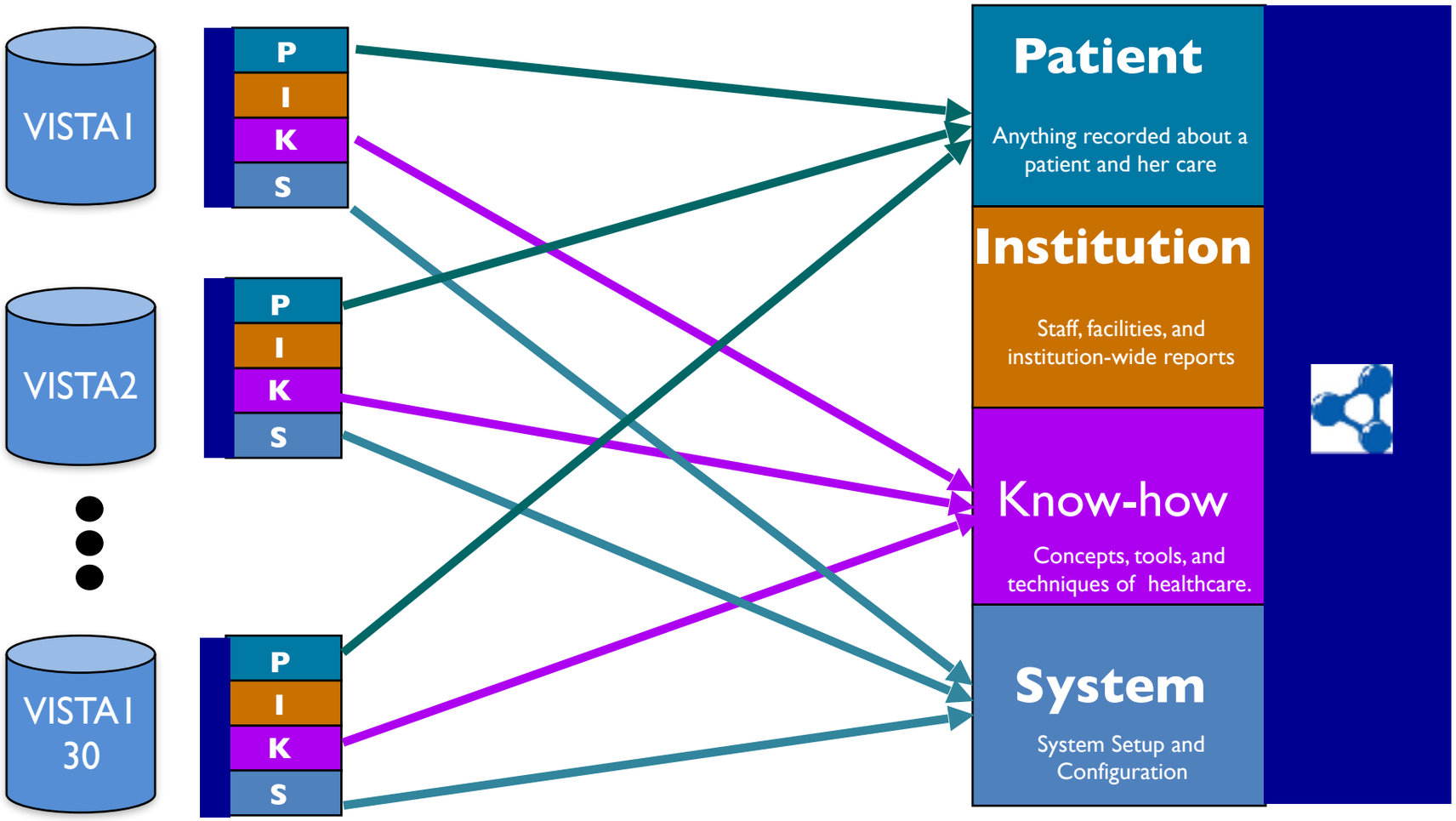
Local VDMs

Analytics

Cross-linking

Linked VISTA (Master VDM)

Enterprise Management



Patient Data Management
Extract and manage patient data with patient-specific security and metadata, allowing patient-centric controls on data access and exchange.

Institutional Data Management:
Institution specific data can be exchangeable and centrally manageable

Knowledge Management:
Common medical concepts, standards, and know-how may be identified and managed as a clearly defined class of VISTA data.

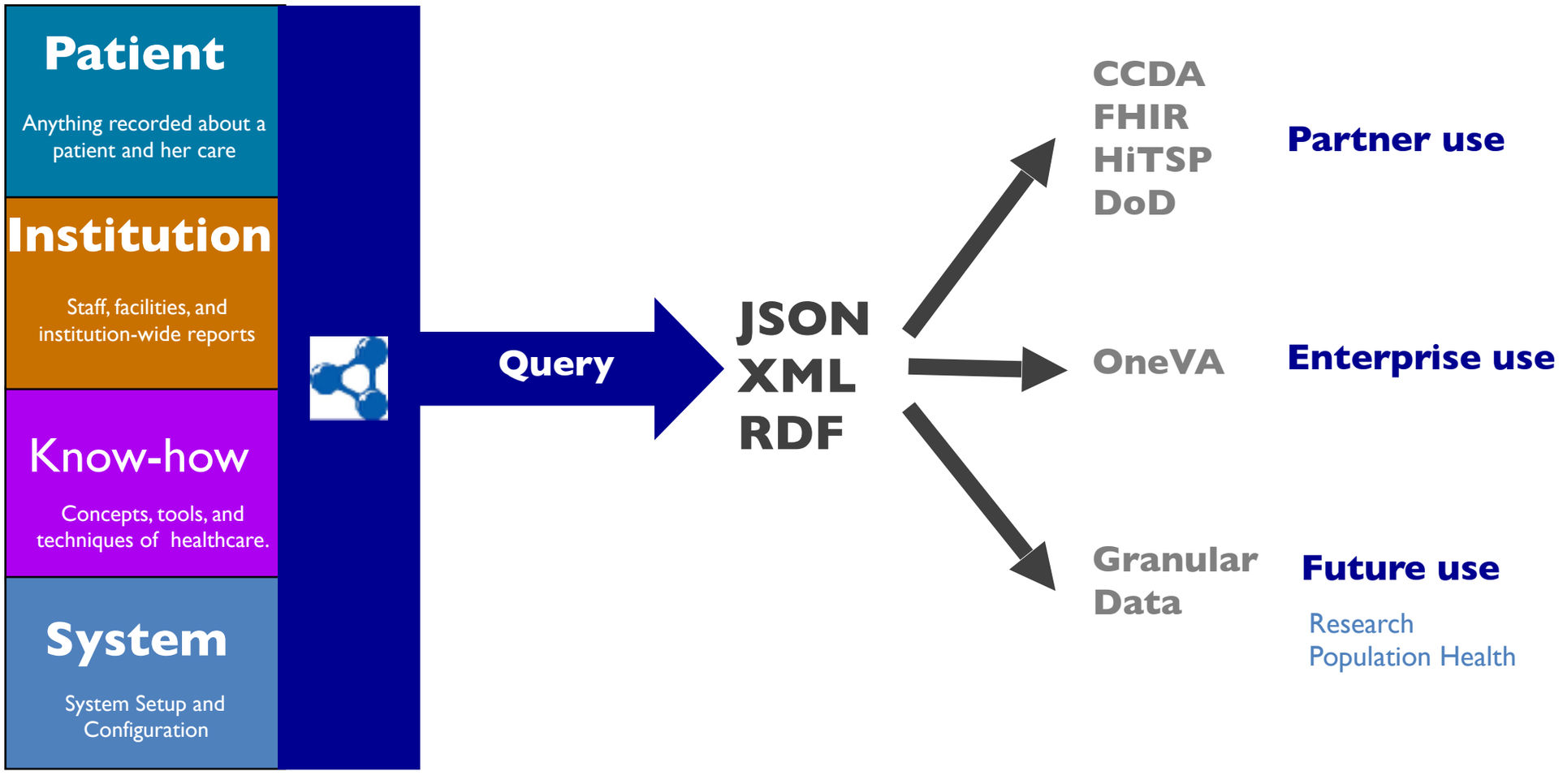
Configuration Management:
The entire configuration of a system can be identified, extracted, and transported, and inserted from system to system

Linked VISTA: Comprehensive Access

One benefit of creating a cross-VISTA enterprise data model (Linked VISTA) is that it can be queried against <u>any</u> VISTA for <u>any</u> data with with <u>one</u> web-standard query.

This would allow any authorized system to securely and directly query authoritative VISTA data in real-time with one standard query language. The output of these queries will be in all modern web-standard forms to maximize secondary use.

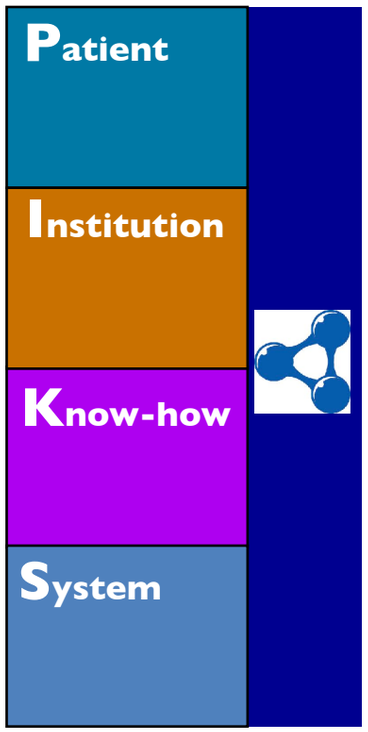
Linked VISTA (VDM)



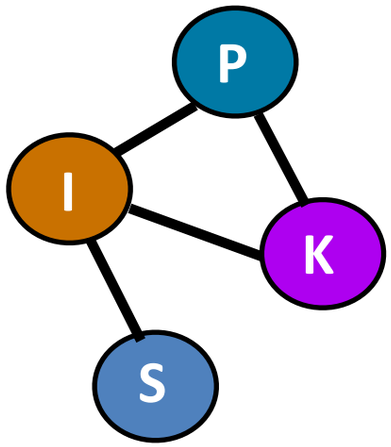
Linked VISTA: ***Web-standard Representation***

Use of web-standard representation maximizes data re-use, meshing, and re-mixing with the maximum number of other sources of data for research and patient care.

Linked VISTA (VDM)



published



Same As

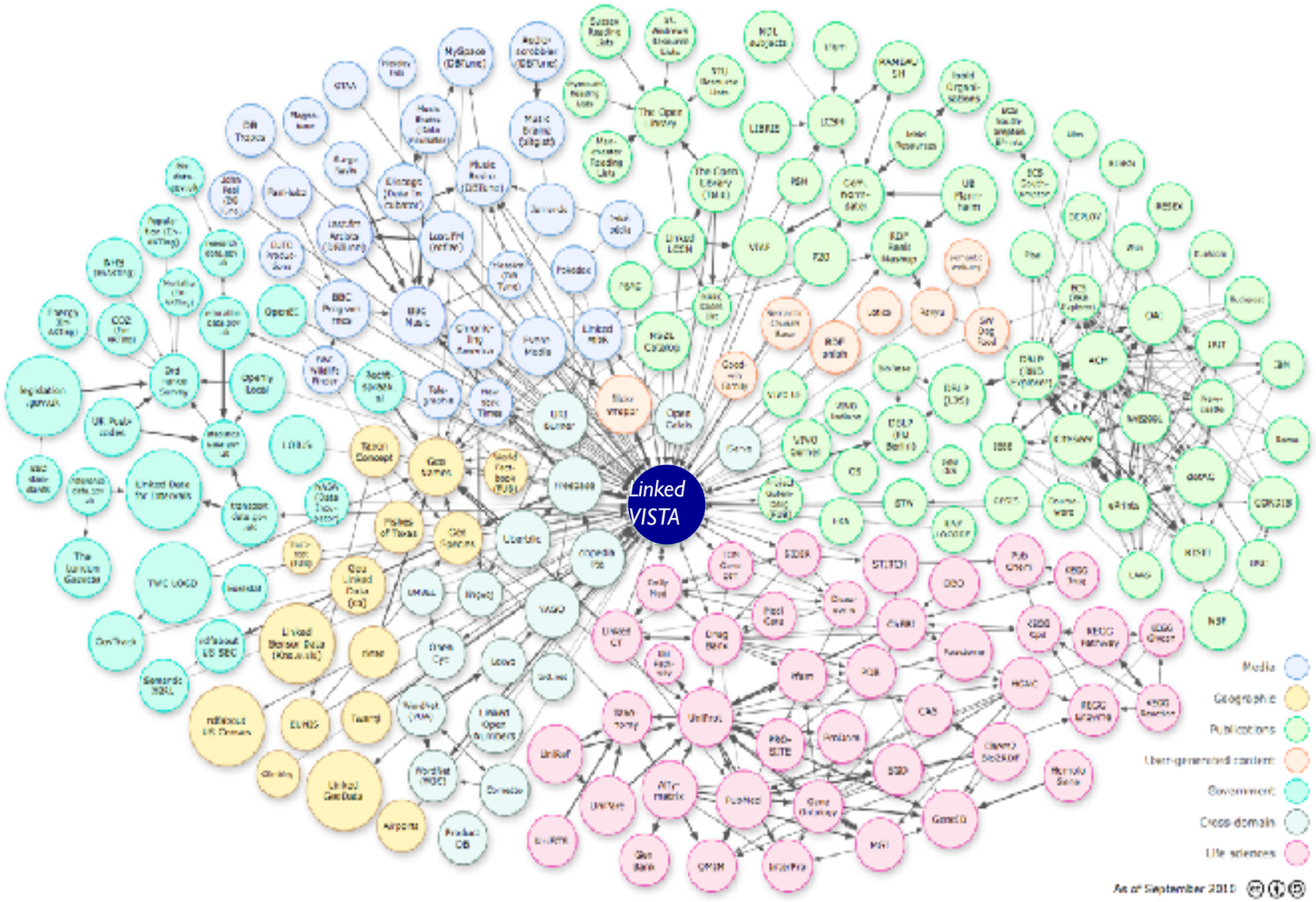
=

Linked Data

Granular
Data-atomic
Meta-data tagged
URI-based
Model-based
Linkable
Web Standard

Linked VISTA: ***Internet-scale semantic integration***

Representing VistA data in a Linked Data form supports real-time semantic integration with thousands of other other linked sources.



Linked VistA

VistA Data - represented in standard Linked Data form - can be interlinked with any and all other Linked Data sources.

This will enable *meshing, enrichment, and augmentation of patient data with all other sources, providing a comprehensive view of all patient data from all locations, clinics, hospitals, or the home.*

Data sources also include patient-generated, mobile device, TeleHealth, and any other Internet-enabled device data (Internet of Things).