

# VISTA Data Project

*Web-scale interfacing for VA's VISTA Data.*

Patient

Provider

Interfaces

# VISTA Data

***35 years of VA's institutional know-how and data comprising hundreds of billions of clinical facts** - and continuing to grow by over a million new lab tests, radiographs, and documents each day - hidden from patients and providers under a sea of interfacing code.*



**VA**  
HEALTH  
CARE

Defining  
**EXCELLENCE**  
in the 21st Century

# ***VISTA Data Project***

## ***A Metadata-Centric Approach to Web-scale integration***

***April 26, 2016***

***Health Information Technology/Informatics Steering Committee  
Defense Health Agency, U.S. Department of Defense  
Fairfax, VA***

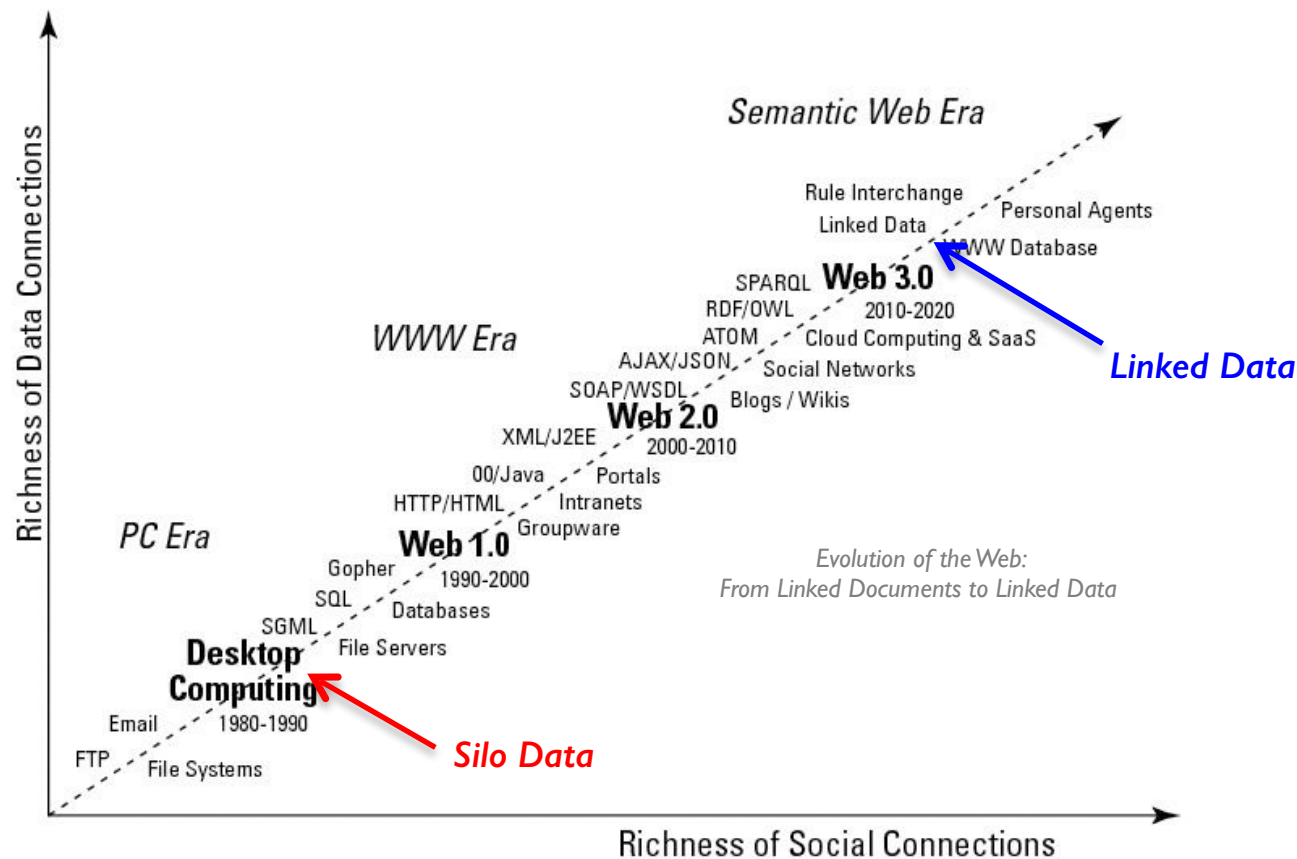
***Rafael Richards MD MS***

***Director, VISTA Data Project***

***Pacific Joint Information Technology Center  
VHA Office of Informatics and Analytics  
U.S. Department of Veterans Affairs***



# Evolution of Data: Increased Connectivity



**VA adopted and deployed VISTA nationwide in 1981 as its national health information system, before the advent of the Internet.** In the pre-internet era (PC Era) databases ran on internal ad-hoc Intranets (Silo Data, red arrow). After the invention of the Internet Protocol in 1982, web technologies rapidly evolved over the past three decades (WWW Era). By adopting the WWW Data model to represent VISTA's native data model (Linked VISTA) - this would allow web-standard, web-scale semantic integration and interfacing of VISTA data across the enterprise and across the Internet.





***Opaque.  
Brittle.  
Inconsistent.  
Insecure.***



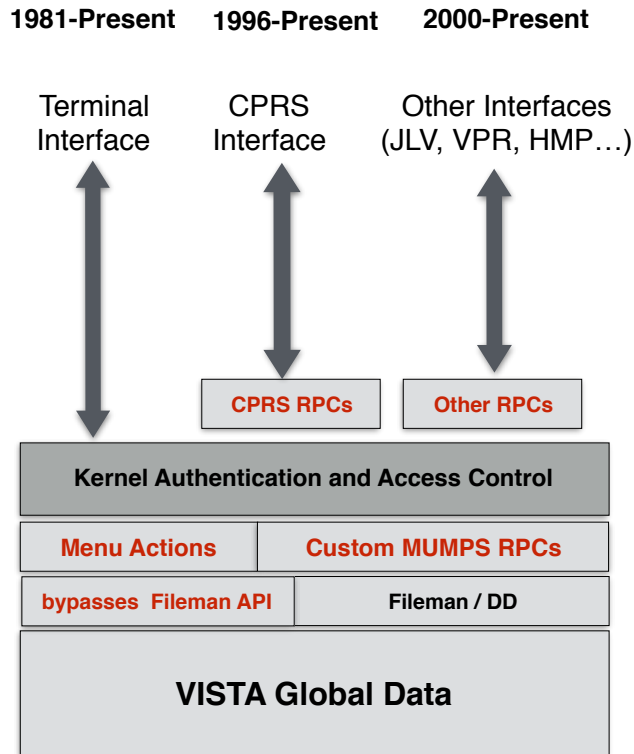
As a result, code-centric interfacing lacks any uniform method to comprehensively or securely interface to VISTA data. There are infinite permutations of hard-coded interfaces possible.





# VISTA Interfacing Evolution

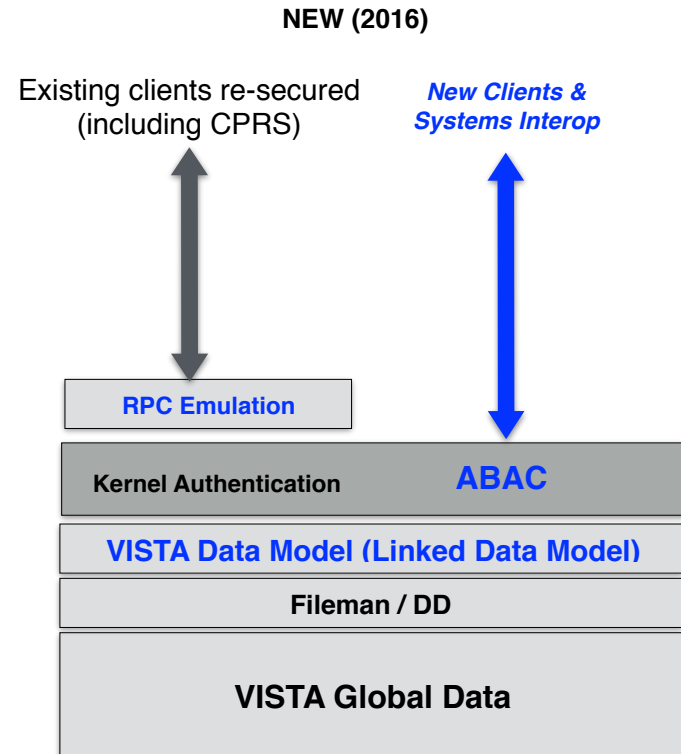
## Code-based Interfaces (x 3500, unique)



Current external interfacing to VISTA is exclusively through MUMPS-based RPCs. These are hard-coded in MUMPS code for specific clients only and not interchangeable to other clients due to embedded business logic within the custom MUMPS and client code. Security for all RPCs is based on the Terminal (roll-and-scroll) interface and its Menu Actions. This is a terminal-only legacy security mechanism, and not applicable to external, Web-, or GUI-based interfaces.

Many of the 3500 RPCs bypass the Fileman API and Data Dictionary, writing direct to MUMPS global storage. Bypassing the FM API means that Fileman security and auditing measures are bypassed, creating a significant security gap. In addition, this makes the data inaccessible to any other applications or by any other method other than by writing yet more custom MUMPS RPCs (The read and write RPCs are completely distinct from each other). The only means to access or interface to new data is to write new MUMPS RPCs using the Terminal-based Actions-centric security, in addition to custom RPC MUMPS security code.

## Model-based Interface (x1, standard)

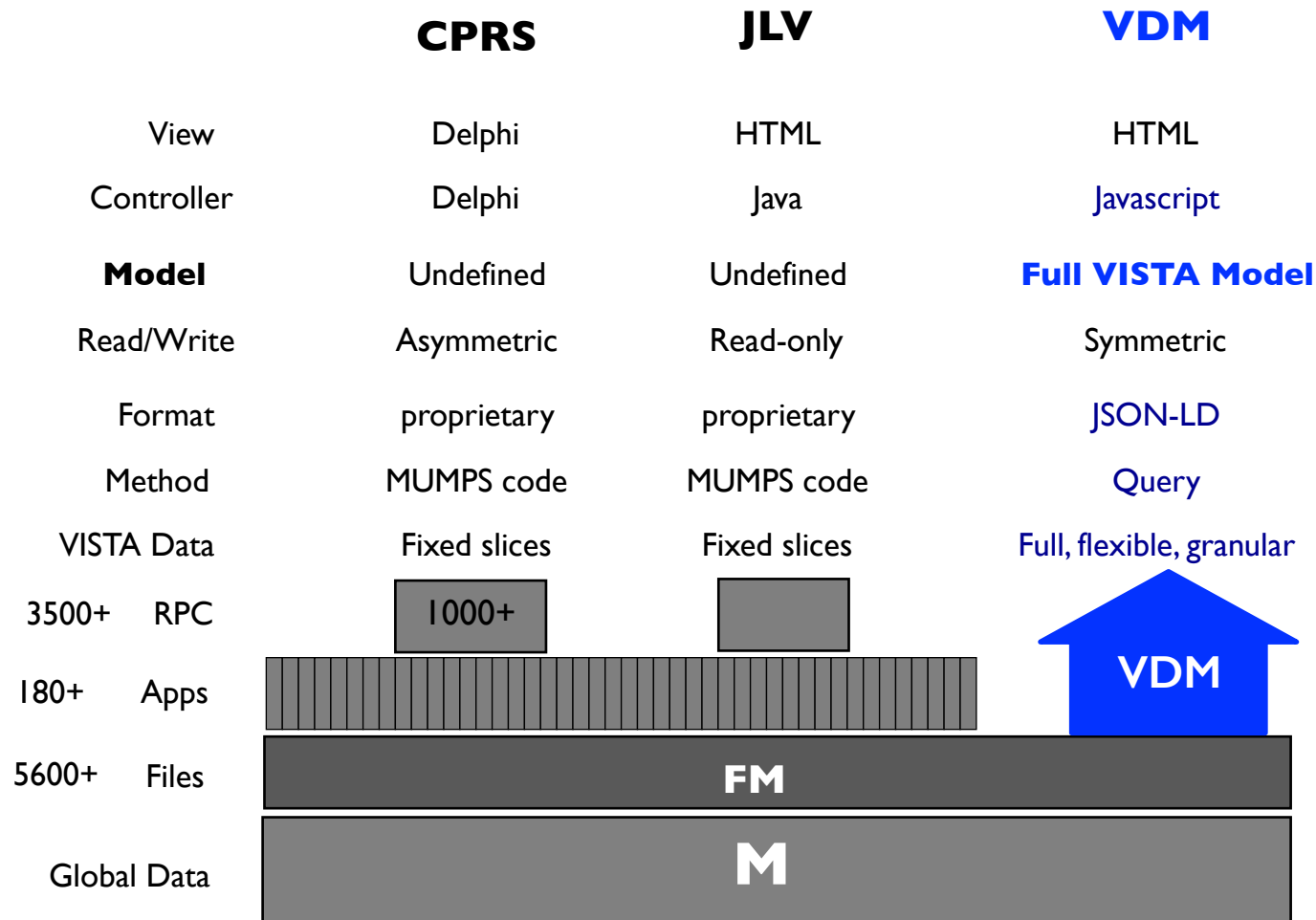


All external interfaces and functionality are Model-driven, language-agnostic, client-agnostic, Fileman API compliant, and secured with both existing Kernel authentication, in addition to new modern, industry-standard, patient-centric, attribute-based access control (ABAC).

All interfacing is through a single, secure, symmetric read-write Master VISTA Data Model using modern, web-standard languages and tools. The read data model is identical to the write data model, making client access simpler. Secure access to all VISTA data is through an ABAC security-enhanced Master VISTA Data Model (MVDM).



# VISTA Interfacing Evolution



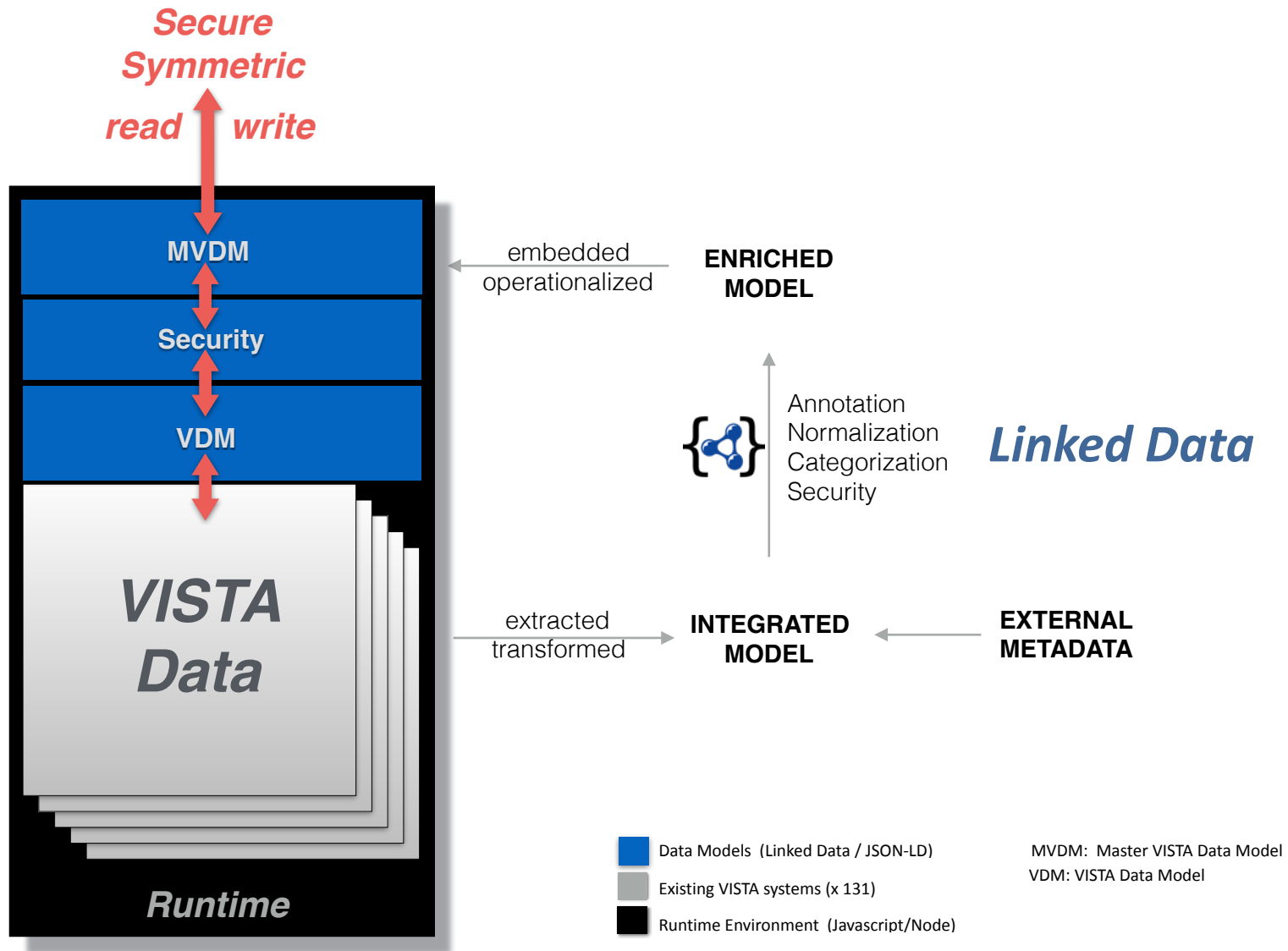
*VISTA Data Model (VDM) can access all data spanning 180+ applications with full granularity and definition because the fully exposed VISTA Data model logically bridges all applications through their native data dictionaries. No legacy APIs, HL7, RPCs, or MUMPS code. Just data. All of it. Defined. Structured.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



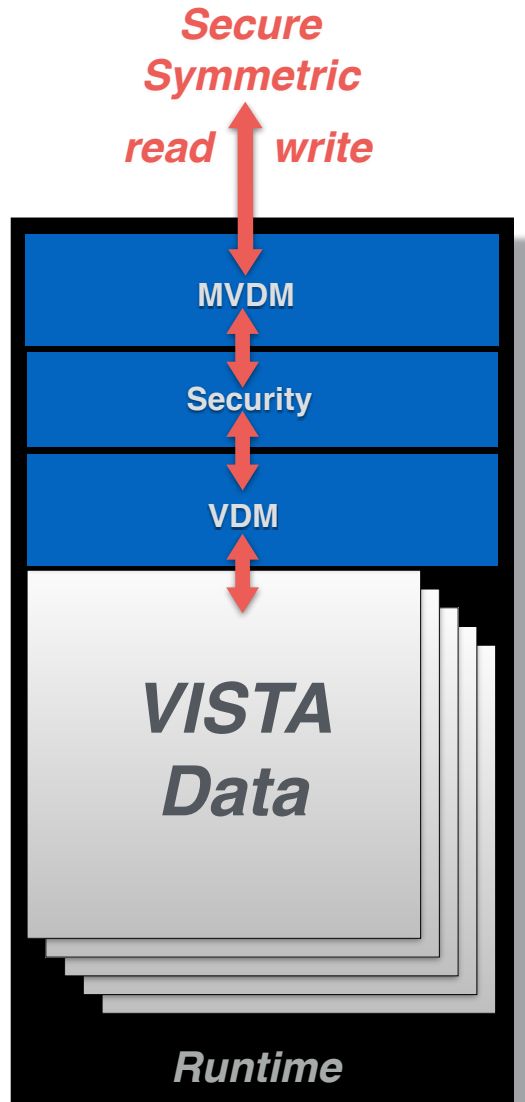
# VISTA Interfacing Evolution

Interface	MUMPS RPCs (x3500)	Master VISTA Data Model (x1)
Method	<ul style="list-style-type: none"> <li>❌ Relies on over 3500 client-specific, non-interchangeable legacy MUMPS routines</li> <li>❌ Distinct, unique routines for reading vs writing the same data</li> <li>❌ Requires extensive knowledge and experience with MUMPS and VISTA</li> </ul>	<ul style="list-style-type: none"> <li>✅ <small>NEW</small> Data Model-Driven</li> <li>✅ <small>NEW</small> Client-agnostic</li> <li>✅ <small>NEW</small> One single, symmetric read-write mechanism for all data.</li> <li>✅ Requires no knowledge or experience with VISTA internals or MUMPS.</li> </ul>
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	<ul style="list-style-type: none"> <li>✅ <small>NEW</small> Applicable to <i>any</i> new interface.</li> <li>✅ <small>NEW</small> Data-Centric;</li> <li>✅ <small>NEW</small> Patient-Centric,</li> <li>✅ <small>NEW</small> Attribute-Based Access Control (ABAC)</li> </ul>
Fileman API Compliant	<ul style="list-style-type: none"> <li>❌ Unreliable, Incomplete</li> <li>❌ Variable compliance</li> </ul>	<ul style="list-style-type: none"> <li>✅ Reliable, Complete</li> <li>✅ 100% Compliant</li> </ul>
Audit	<ul style="list-style-type: none"> <li>❌ Incomplete</li> <li>❌ Bypassess Fileman auditing</li> </ul>	<ul style="list-style-type: none"> <li>✅ Comprehensive AND</li> <li>✅ <small>NEW</small> Patient-Centric</li> </ul>
Unit Tested	<ul style="list-style-type: none"> <li>❌ NO</li> <li>❌ 0% logic tested</li> </ul>	<ul style="list-style-type: none"> <li>✅ YES</li> <li>✅ 100% logic validated</li> </ul>
Documentation	<ul style="list-style-type: none"> <li>❌ Incomplete, inconsistent, unclear.</li> <li>❌ Requires understanding MUMPS code</li> </ul>	<ul style="list-style-type: none"> <li>✅ Complete, consistent, clear.</li> <li>✅ <small>NEW</small> Core is machine generated</li> </ul>

# 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)

# VISTA Data Model Features

VISTA Data	Details
 Access	<p><b>A single, universal, industry-standard mechanism for reading and writing <i>all VISTA data</i>.</b></p> <p>This mechanism is unified through a read model and write model integrated into a single, symmetric-read-write data model (VDM), with all data in industry-standard web formats. <i>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.*</i></p>
 Integrity	<p><b>Comprehensive, automated, standardized, strict data integrity enforcement for <i>all VISTA data</i>.</b></p> <p><i>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: <a href="#">Master Data Management</a></i></p>
 Security	<p><b>Comprehensive, industry-standard, fine-grained, data-centric security for <i>all VISTA data</i>.</b></p> <p>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 <i>what categories of data it is managing</i> 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 <a href="#">Data-Centric Security</a>, <a href="#">Logical Security</a>, <a href="#">Semantic Security</a> and <a href="#">Attribute-Based Access Control (ABAC)</a></p>

# VISTA Data Model Attributes

<b>Representative</b>	<p><b>VDM operationalizes all relevant VA VISTA data to the maximum extent available.</b></p> <p>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.</p>
<b>Real-Time</b>	<p><b>VDM is operationalized using Best-of-Breed real-time server-side runtime technology.</b></p> <p>The same runtime technology that runs the largest commercial real-time high-traffic websites such as Walmart, eBay, PayPal, Netflix, Uber, LinkedIn, and the New York Times also runs MVDM. <i>This maximizes transactional processing performance directly on the transactional database.</i></p>
<b>Noninvasive</b>	<p><b>VDM provides VISTA with essential new functionality within the current VISTA architecture 'as is', without modification.</b></p> <p>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.</p>
<b>Self-Contained</b>	<p><b>VDM runs entirely server-side, embedded directly on the existing VISTA database.</b></p> <p>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. <i>This makes it easy to deploy, maintain, and keep highly performant. No moving parts. No external dependencies. No middleware.</i></p>

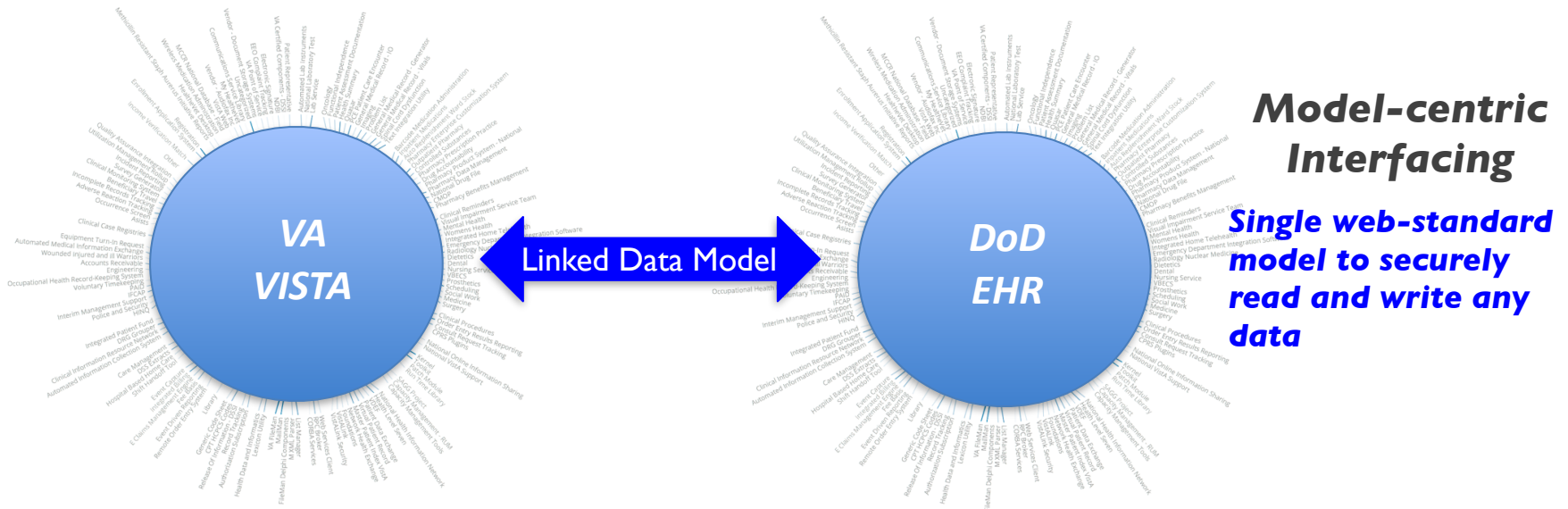
# VISTA Data Model Attributes (continued)

<b>Self-Contained</b>	<p><b>VDM runs entirely server-side, embedded directly on the existing VISTA database.</b></p> <p>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. <i>This makes it easy to deploy, maintain, and keep highly performant. No moving parts. No external dependencies. No middleware.</i></p>
<b>Data-Centric</b>	<p><b>VDM is a completely new, purely data-centric approach to managing VISTA's data.</b> 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 <b><i>comprehensively exposes all the data, which exposes the fact that VISTA has a data model</i></b> - which up to this point has not been realized nor taken advantage of. <i>This is the opposite of a code-centric approach, which obfuscates the data and its data model.</i></p>
<b>Web-Standard</b>	<p><b>VDM technologies are 100% web standard</b> and all used in production settings by the worlds' largest corporations and organizations. For further information see <a href="#">standards and technologies</a>.</p>
<b>Empiric Evolution</b>	<p><b>VDM employs a new approach to empirically evolving VISTA's capabilities through rapid, iterative, functional prototypes.</b> 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 <i>the opposite waterfall development</i>. See <a href="#">spiral model</a></p>





***Comprehensive. Computable. Standard. Secure.***



**Representing the VA and DoD health information systems as Linked Data Model would support computable semantic interfacing and integration between federal agencies and the private sector using a web-standard, web-scale model.**



# ***Reference***

*Web:* [\*vistadataproject.info\*](http://vistadataproject.info)

*Github:* [\*github.com/vistadataproject\*](https://github.com/vistadataproject)

*Email:* [\*rafael.richards@va.gov\*](mailto:rafael.richards@va.gov)