C3.ai

IT for Enterprise AI

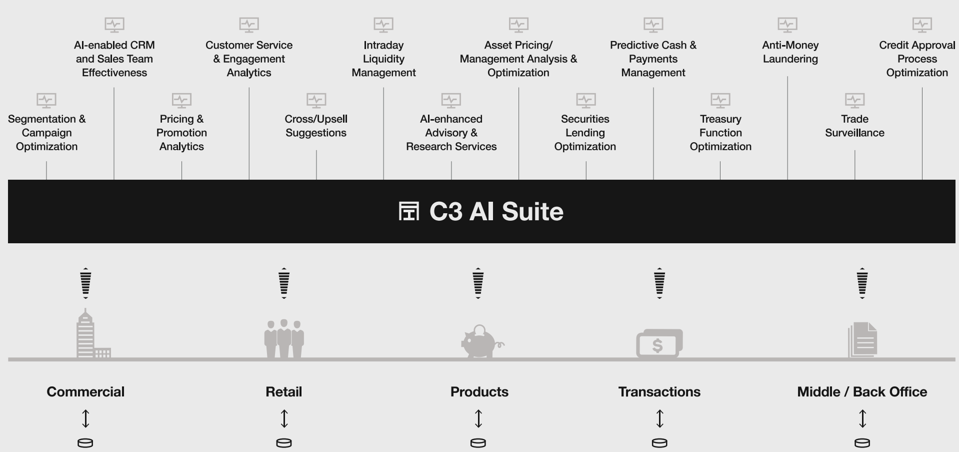
Next-Generation Data Strategies and Platforms for AI at Enterprise Scale

Introduction

Significant opportunity for innovation and competitive advantage lies in applying AI to re-think how businesses operate and deliver dramatic improvements in how companies engage customers, make better use of their workforce and improve business operations. Business process re-engineering leveraging ubiquitous computing, data from corporate systems, the proliferation of sensors and the internet, and learning algorithms is commonly referred to as Digital Transformation. Yet today, prevalent use of AI in business is limited.

The use cases for AI in banking, for instance, are numerous. For example, AI applied to data produced by transaction and order systems, product systems, client masters, and document repositories can proactively identify the need to address corporate cash churn or to prioritize anti-money laundering efforts. AI and optimization techniques can be used to anticipate fluctuations in customer demand or supply disruptions, to better inform securities lending efforts, or for early identification of loan application risk. These are just a few of the many high-value AI use cases in the banking sector. Similarly, AI has been proven to significantly improve a wide range of processes across multiple industries, including manufacturing, aerospace, oil and gas, defense, healthcare, and utilities.

Representative Value Chain: Banking



The information technology challenges to delivering these AI applications are daunting.

The baseline capability required is aggregation and processing of rapidly growing petabyte-scale datasets continuously harvested from thousands of disparate legacy IT systems, internet sources, and multi-million sensor networks. In the case of one Fortune 500 manufacturer, the magnitude of the data aggregation problem is 50 petabytes fragmented across 5,000 systems representing customer, dealer, claims, ordering, pricing, product design, engineering, planning, manufacturing, control systems, accounting, human resources, logistics, and supplier systems fragmented by mergers and acquisitions, product lines, geographies and customer engagement channels (i.e., online, stores, call center, field).

Next Generation Application Platform Technical Requirements

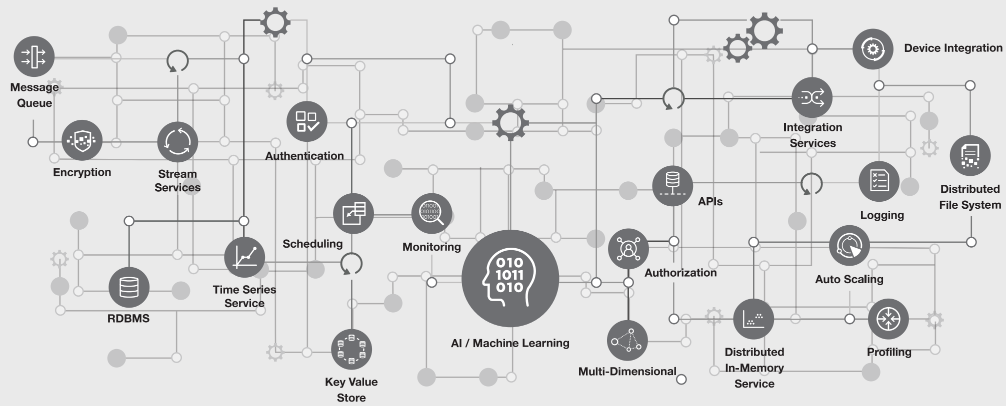
The complete technical requirements for an AI application platform include:

1. **Data Aggregation** from enterprise and extraprise systems, and sensor networks
2. **Multi-Cloud Computing** for cost effective elastic scale-out compute and storage on private and public clouds
3. **Edge Computing** for low-latency local processing and AI predictions/inferences
4. **Platform Services** for continuous data processing, temporal and spatial processing, security and data persistence
5. **Enterprise Semantic Model** to provide a consistent object model across a business
6. **Enterprise Microservices** to provide a catalog of AI based software services
7. **Enterprise Data Security** provides robust user access authentication and authorization
8. **System Simulation** using AI and Dynamic Optimization Algorithms with full lifecycle support including development, testing, and deployment
9. **Open Platform** with support for multiple programming languages, standards-based interfaces (APIs), open source machine learning and deep learning libraries, and third-party data visualization tools
10. **Common Platform** for Collaborative Development between Software Developers and Data Scientists

Cloud Vendor Tools and Structured Programming

A number of enterprises have attempted to address these requirements by assembling various native services and microservices offered by public cloud providers – e.g., AWS, Azure, and others – to build Enterprise AI applications. Leading vendors like AWS are developing a growing set of services and microservices, each of which provides useful functionality to address various Enterprise AI requirements. This requires the use of traditional "structured programming" to stitch together the various cloud services into a working application. While elastic cloud platforms such as AWS and Azure provide rich infrastructure-level services that are essential for Enterprise AI, this approach has proven ineffective for rapidly developing, deploying, and scaling Enterprise AI applications in a repeatable manner.

Figure: Structured programming results in complex code that is brittle and difficult and costly to maintain.



The problem with this approach is its high level of complexity: Because these systems lack a model-driven architecture like that described in the following section, developers need to employ structured programming to stitch together the various cloud services – resulting in numerous component interdependencies and the need to write, test, and debug many lines of code, creating brittle applications that are difficult and costly to maintain. Using traditional structured programming, the number of permutations of infrastructure service calls, enterprise systems and data integrations, enterprise data objects, sensor interfaces, application and data science programming languages and libraries to support application development is almost infinite.

Model-Driven Architecture

The architecture requirements for a distributed AI platform are uniquely addressed through a Model-Driven Architecture. This architecture abstracts application and machine learning code from the underlying platform services and provides a domain-specific language (annotations) to support highly declarative, low code application development.

WIKIPEDIA
The Free Encyclopedia

The model-driven architecture approach defines system functionality using a platform independent model (PIM) using an appropriate domain-specific language (DSL).

Then, given a platform model, the PIM is translated to one or more platform-specific models (PSMs) that computers can run. This requires mappings and transformations and should be modeled too.

The PSM may use different DSLs or general purpose language. Automated tools generally perform this translation.

The model-driven approach provides an abstraction from the underlying technical services (for example, queuing services, streaming services, ETL services, data encryption, data persistence, authorization, authentication) and simplifies the programming interface required to develop AI apps to a Type System interface.

The model is used to represent all layers of an application including the data interchange with source systems, application objects and their methods, AI-machine learning algorithms and the application user interface. Each of these layers are also accessible as microservices.

C3.ai Type System

The C3.ai Type System is a data object-centric abstraction layer that binds the various C3 AI Suite components, including infrastructure and services. It is both sufficient and necessary for developing and operating complex predictive analytics applications in the cloud.

The C3.ai Type System is the medium through which application developers and data scientists access the C3 AI Suite, C3.ai Data Lake, C3.ai Applications, and applications and microservices. Examples of C3.ai Types include data objects (e.g., customer, product, supplier, contract, or sales opportunity) and their methods, application logic, and machine learning classifiers.

The C3.ai Type System allows programs, algorithms, and data structures – written in different programming languages, with different computational models, making different assumptions about the underlying infrastructure –to interoperate without knowledge of the underlying physical data models, data federation and storage models, interrelationships, dependencies, or the bindings between the various structural platform or cloud infrastructure services and components (e.g., RDBMS, No SQL, ETL, SPARK, Kafka, SQS, Kinesis, object models, classifiers, data science tools, etc.). The C3.ai Type System provides RESTful interfaces and programming language bindings to ALL underlying data and functionality.

Leveraging the C3.ai Type System, application developers and data scientists can focus on delivering immediate value, without the need to learn, integrate, or understand the complexities of the underlying systems. The C3.ai Type System enables programmers and data scientists to develop and deploy production AI, big data, and predictive analytics applications in one-tenth the time at one-tenth the cost of alternative technologies.

To improve manageability, Types support multiple object inheritance (allowing objects to inherit characteristics from one or more other objects). For example, a building might have characteristics of both a residential and commercial use building.

The Type system, through inherent dataflow capabilities, automatically triggers the appropriate processing of data changes by tracing implicit dependencies between objects, aggregates, analytic features and machine learning algorithms in a directed acyclic graph.

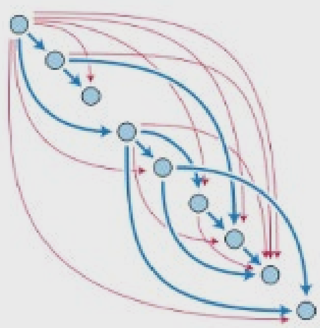


Figure: Data changes automatically trigger data pipelines including re-computation of analytic features, aggregates and AI predictions

The Type System is accessible through multiple programming language bindings (i.e., Java, JavaScript, Python, Scala, and R), and Types are automatically accessible through RESTful interfaces allowing interoperability with external systems.

AI & Optimization to Inform Simulation

The Type System supports combining AI and optimization algorithms to tackle challenging system simulation problems requiring dynamic probabilistic forecasts and constraint programming.

For example, C3.ai Inventory Optimization uses advanced AI and stochastic optimization techniques to account for both supply and demand-side uncertainties. C3.ai Inventory Optimization and the C3 AI Suite dynamically optimize inventory levels for individual parts on a near-real time basis to reduce excess inventory, maintain SLAs, and minimize the likelihood of stock-outs. This differs significantly from alternative approaches that account for stochasticity through static Monte Carlo methods, rather than the AI-based self-learning approach C3 AI Suite offers to better account for continuously changing circumstances.

Figure: Model-Driven Architecture Abstracts Underlying Platform Services through a simple Type Systems Interface

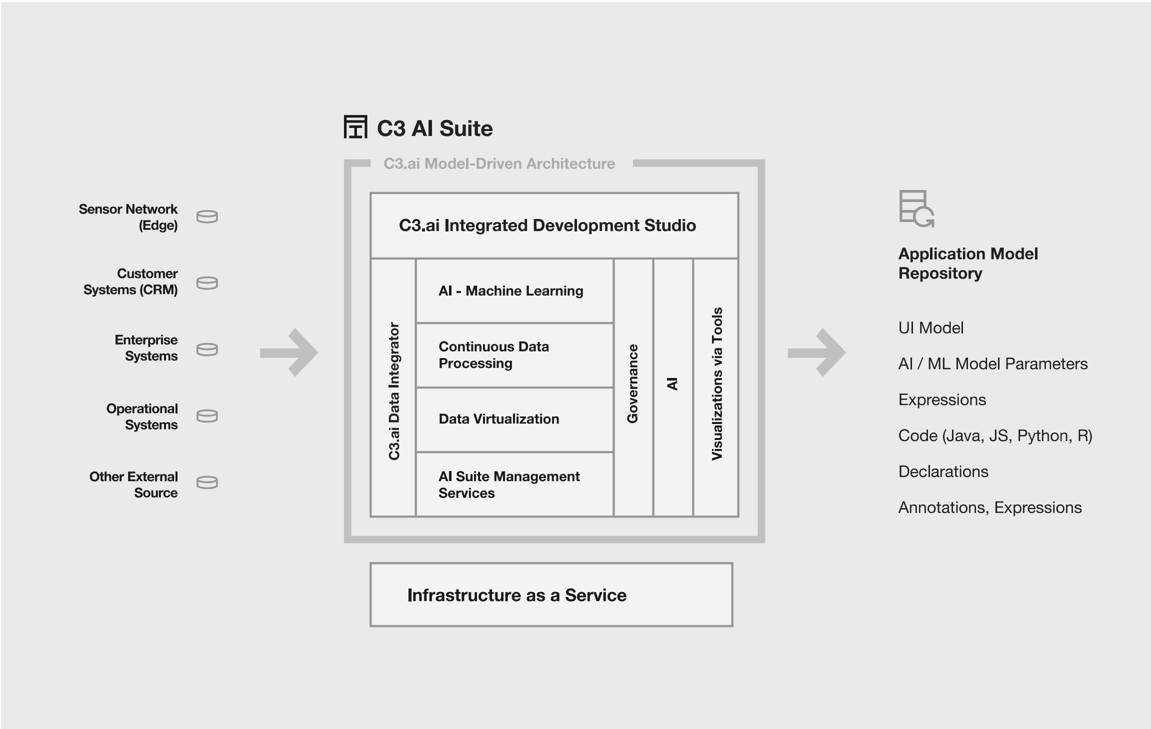
Modeling AI Applications

C3.ai applications are expressed through the following model artifacts:

1. **Metadata** – example - used to declare objects and their attributes
2. **Annotations –** example - used to indicate the underlying data store to persist an object's data
3. **Expressions –** example - used to represent functions which perform mathematical operations on input data without programming logic
4. **Programming Logic** – example - used when necessary to express logic beyond metadata, annotations and expressions

These artifacts are used to represent the following interdependent "layers" of an application:

1. **Data Exchange Object Model, Mappings, and Transformations –** metadata representing a data model optimized/denormalized for data interchange between systems, mappings and data transformations to target an application object model.
2. **Application Object Model –** metadata representing a data model optimized for development of applications and machine learning models.
3. **Methods –** business logic representing actions (e.g., enroll a customer, maintain equipment) on objects expressed in code
4. **Analytic Features –** expressions which perform spatial and/or temporal mathematical operations on input data
5. **AI Machine Learning Model –** combination of analytic features and AI-ML algorithm (e.g., NLP pipeline, logistic regression, tensor flow…)
6. **Security Services –** annotations and expressions applied to Objects, Methods, Features and Machine Learning Models.
7. **User Interface (Optional)** – metadata representing user interface fields, their bindings to objects and object properties, and user interface objects (charts, grids, etc.,)



Model artifacts are extensible and upgradeable:

Model Extensibility

- Meta Data Extensions
- Code Extensions

Model Upgradeability

- Platform Upgrades
- Application Upgrades
- Preserving Customer Extensions

The benefits of a model-driven architecture include: 1) future proofing investments in application and microservices code as underlying infrastructure services rapidly evolve; 2) less app code required to be written, quality assured, and maintained resulting in significantly faster application development and lower total cost of ownership; and 3) upgradeability.

Common Platform for Application Developers and Data Scientists to Collaboratively Develop AI Applications

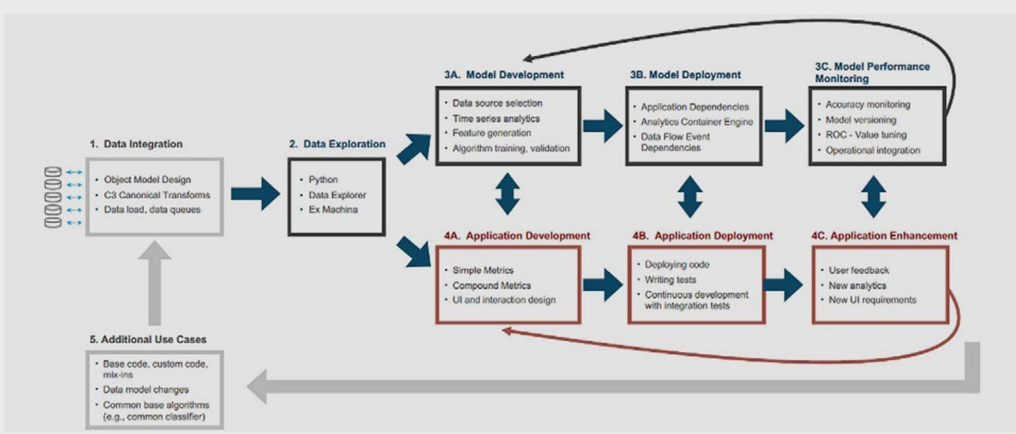
Data scientists typically work in isolation, developing and testing machine learning algorithms against small subsets of data provided by IT from one or more disparate source systems. The bulk of their time is spent on data cleansing and data normalization to represent the same entities, measures (units), states (e.g. status codes) and events consistently in time and across systems and to correlate ("join") data across systems. The resulting algorithms, typically written in Python or R, do not conform to IT standards and require rewriting to different programming language such as a Java. Furthermore, the efficacy of the algorithm is sub-optimal since it has not been tuned against a representative production data set.

A platform is required to allow data scientists to develop, test, and tune algorithms in the programming language of their choice against a snapshot of all available production data. To accelerate development, data scientists can leverage work completed on the platform by data engineers and application developers to handle data cleansing, data normalization, object modeling and representation, microservices to focus on analytic feature development for classic machine learning or deep learning models.

The resulting machine learning algorithm is immediately deployable in production and available as a microservice through a standard RESTful API.

Rapid App Development and Data Science Life Cycle on the C3 AI Suite

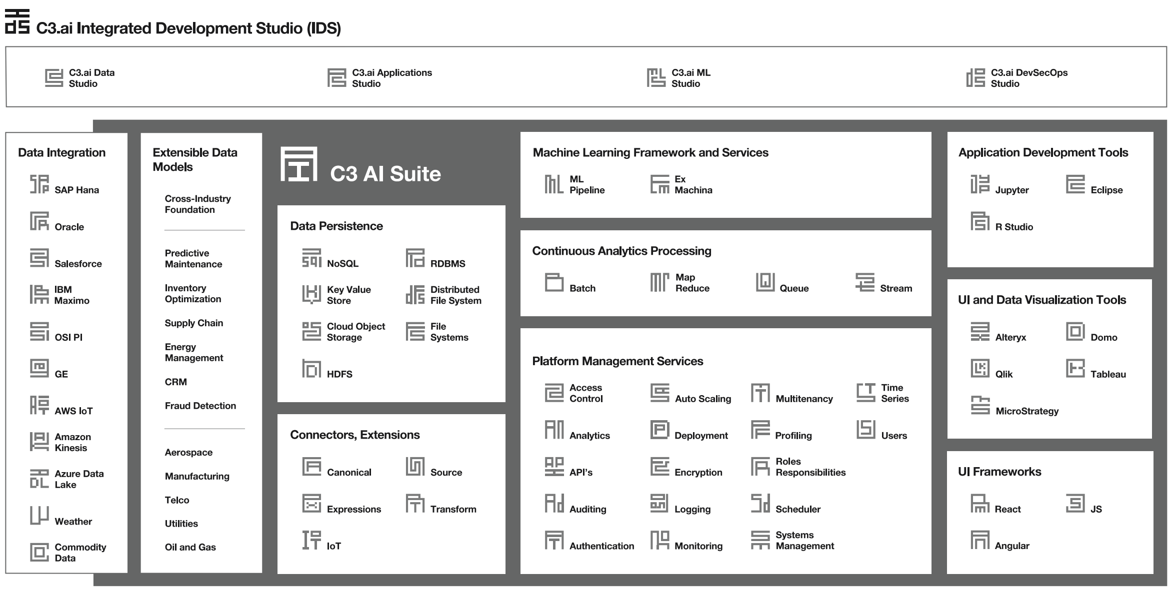
Figure: A next-gen Application Platform supports collaborative iterative development between Application Developers and Data Scientists. Each team takes advantage of the work performed by the other even though they may be using different programming languages



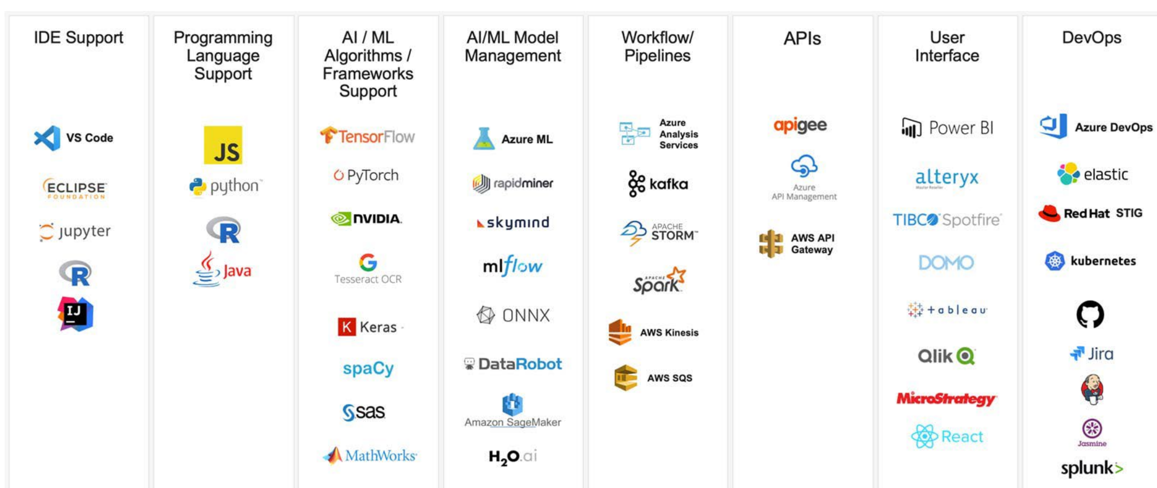
The C3 AI Suite is an example of a proven model-driven AI platform.

The C3 AI Suite allows small teams of five to ten application developers and data scientists to collaboratively develop, test and deploy large-scale production AI applications in one to three months. The platform is proven across thirty large scale deployments across industries including Energy, Manufacturing, Aerospace / Defense, Healthcare, and Financial Services. A representative large scale C3 AI Suite deployment processes AI inferences at a rate of a million messages per second against a petabyte-sized unified federated cloud data image aggregated from fifteen disparate corporate systems and a forty million sensor network. Global 1000 organizations have successfully used the platform to deploy full scale production deployments in 6 months and enterprise-wide digital transformations with over twenty AI applications in 24 to 36-month timeframes.

C3 AI Suite for Digital Transformation



The C3 AI Suite is also an open platform with plug-ins and flexibility for data scientists and developers – including IDEs and tools, programming languages, tools, DevOps capabilities, and others.



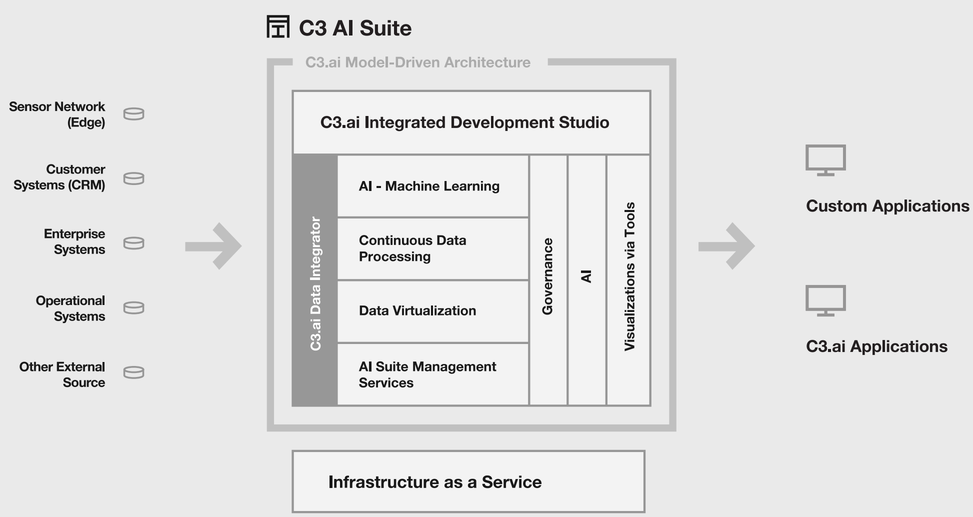
This section reviews each technical requirement and how they are satisfied through components of a next-generation application platform.

1. Data Aggregation: Unified Federated Data Image Across the Business

Process re-engineering across a company's business requires integrating data from numerous systems and sensors into a unified federated data image, and keeping that data image current in near real-time as data changes occur.

To facilitate data integration and correlation across systems, the C3 AI Suite provides a Data Integration service with a scalable enterprise message bus. The Data Integration service provides extensible industry-specific data exchange models. Examples of data exchange models include HL7 for healthcare, eTOM for telecommunications, CIM for power utilities, PRODML and WITSML for oil & gas, and SWIFT for banking. Mapping source data systems to a common data exchange model significantly reduces the number of system interfaces required to be developed and maintained across systems. As a result, C3 AI Suite deployments with integrations to 15 to 20 source systems typically take 3–6 months as opposed to years.

Unified Federated Data Image



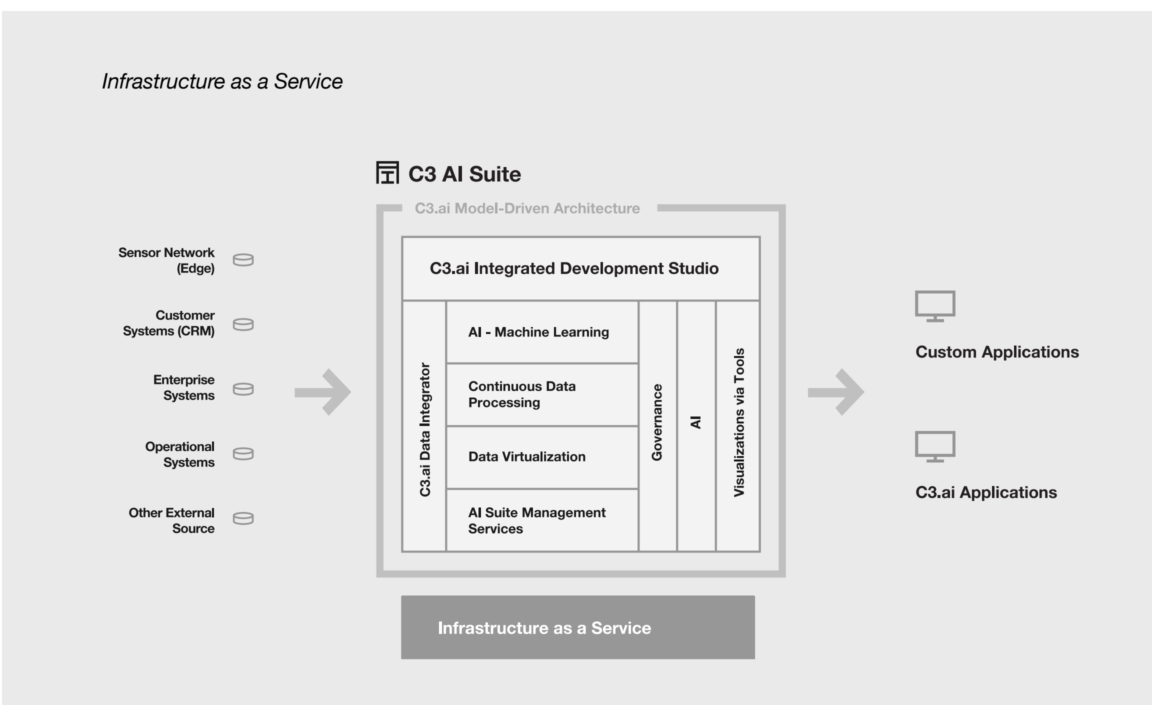
2. Multi-Cloud Computing and Data Persistence (Enterprise Data Lake)

Cost effectively processing and persisting large scale datasets require an elastic cloud scale-out/in architecture, with support for private cloud, public cloud, or hybrid cloud deployments. Cloud portability is achieved through container technology (for example, Mesosphere). However, the platform is optimized to take advantage of differentiated services. For example, the platform takes advantage of AWS Kinesis and Azure Streams when running on AWS and Azure respectively.

Multi-Cloud operation is also supported. For example, the Platform Services can operate on AWS and invoke Google Translate or speech recognition services and access data stored on a private cloud. An instance of the Platform might need to be deployed in country on Azure Stack to conform to data sovereignty regulations. The platform should support installation in a customer's virtual private cloud account (e.g., Azure or AWS account) and support for deployment in specialized clouds such as AWS GovCloud or C2S with industry or government specific security certifications.

Support for running analytics and AI predictions (or inferences) on remote gateways and edge compute devices is also required to address low-latency compute requirements or in situations where network bandwidth is constrained or intermittent (e.g., aircraft).

Data persistence of the unified data image requires a multiplicity of data stores depending on the data and anticipated access patterns. Relational databases are required to support transactions and complex queries, and key-value stores for data such as telemetry requiring low-latency reads and writes. Other stores, including distributed file systems such as HDFS, are required for support of unstructured audio or video, multi-dimensional stores, and graph stores. If a Data Lake already exists, the platform can map and access data in-place from that source system.

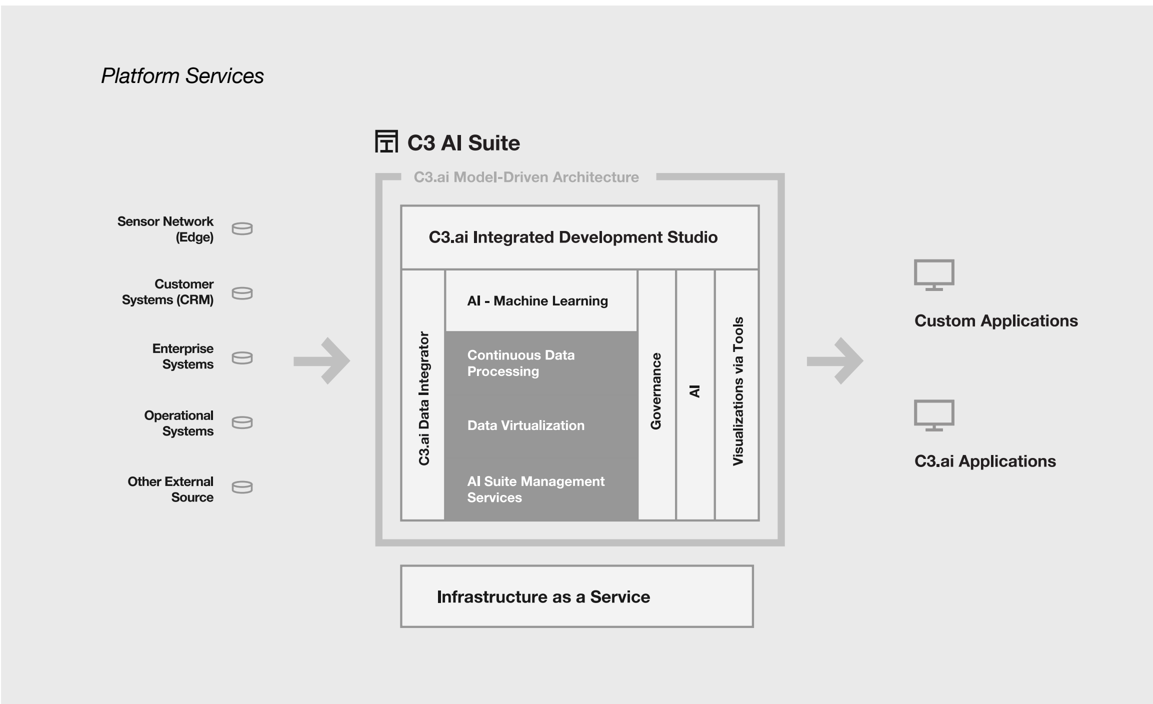


3. Platform Services & Data Virtualization for Accessing Data In-Place

AI applications require a comprehensive set of platform services for processing data in batches, microbatches, real-time stream data and iteratively in memory across a cluster of servers to support data scientists testing analytic features and algorithms against production scale data sets. Secure data handling is also required to ensure data is encrypted while in motion or at rest.

The model-driven architecture, through the Type System, abstracts these native platform services to allow pluggability of these services without the need to alter application code.

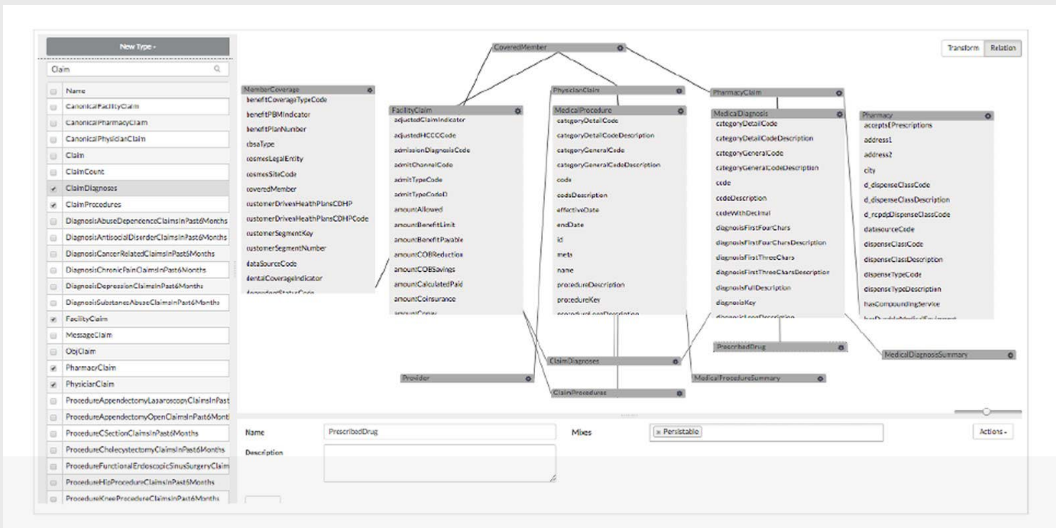
The model-driven architecture supports data virtualization, which allows application developers to instantiate Types and manipulate data without knowledge of the underlying data stores. The platform provides support for relational stores, distributed file systems, key-value stores, graph stores as well as legacy applications and systems such as SAP, OSIsoft PI, and SCADA systems.



4. Enterprise Semantic Model and Microservices

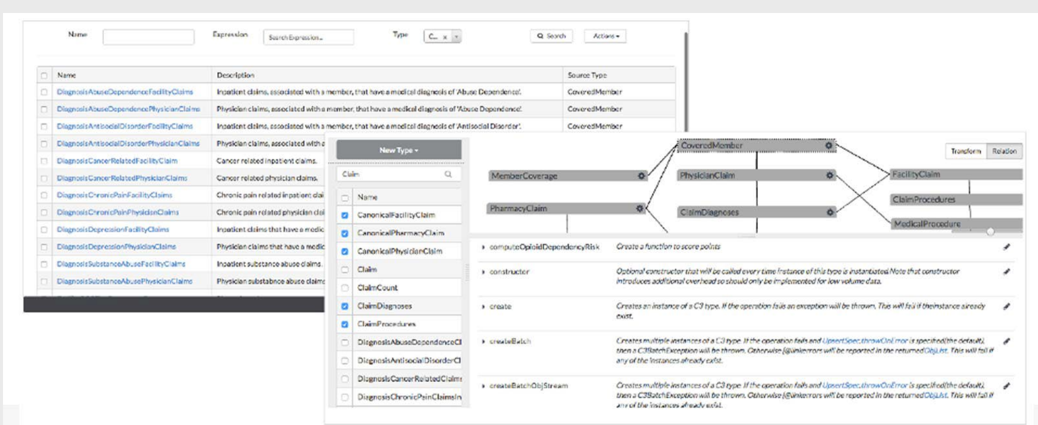
Re-engineering processes across an organization, requires a consistent semantic (object) model across the enterprise. The semantic model represents objects and their relationships independent of the underlying persistence data formats and stores. In contrast to passive entity / object models in typical modeling tools, the object model is active and interpreted by the platform at runtime providing significant flexibility to handle object model and schema changes. Changes to the object model are versioned and immediately active without need to re-write application code.

Enterprise Semantic Model



Similarly, a catalog of AI microservices is published and available enterprise-wide subject to security / authorization access controls.

Enterprise AI Microservices



5. AI & Optimization

An integrated full life-cycle algorithm development experience is required for data scientists to rapidly design, develop, test and deploy machine learning and deep learning algorithms.

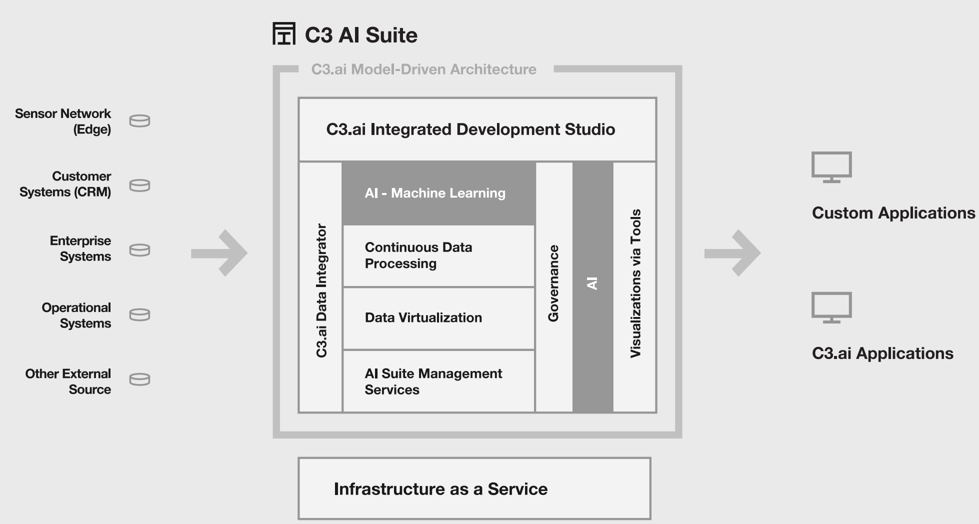
This allows data scientists to use the programming language of their choice – Python, R, Scala, Java – to develop and test machine learning and deep learning algorithms against a current production snapshot of all available data. This ensures the data scientists achieve the highest levels of machine learning accuracy (precision and recall).

The machine learning algorithm should be deployable in production without the effort, time, and errors introduced by translation to a different programming language.

The AI machine learning algorithms should provide APIs to programmatically trigger predictions and re-training as necessary.

AI predictions should be conditionally triggered based on the arrival of dependent data. AI predictions can trigger events and notifications or be inputs to other routines including simulations involving constraint programming.

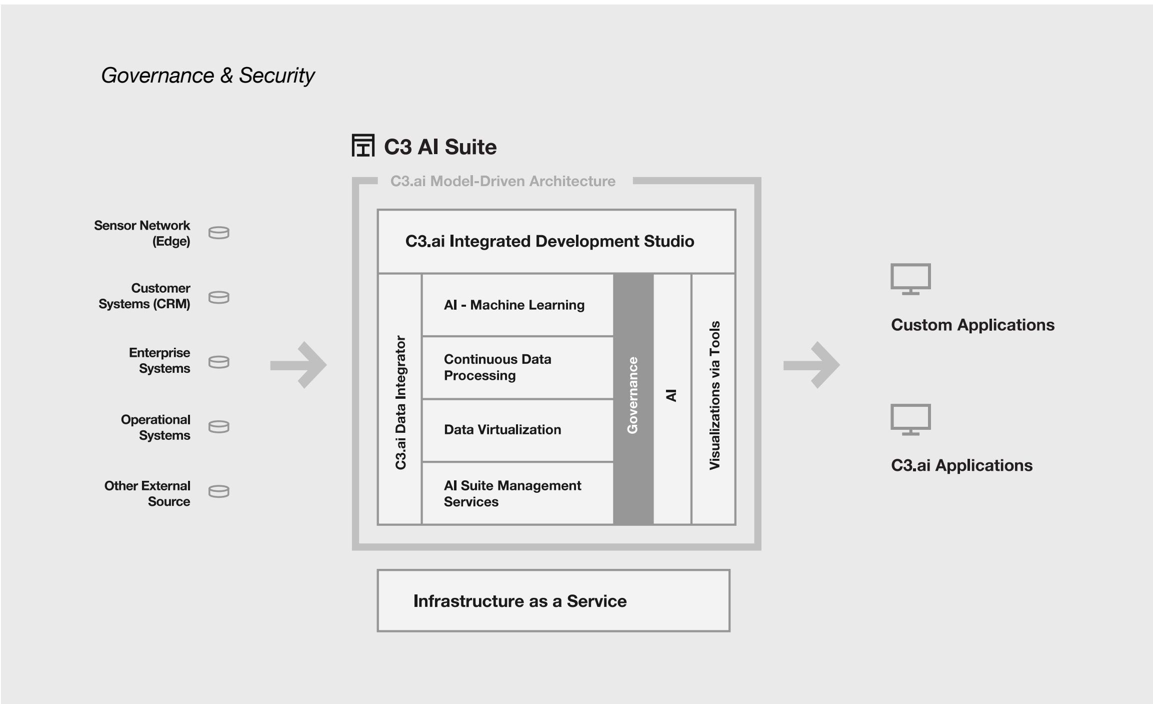
AI Architecture



6. Data Governance & Security

The platform should support multi-level user authentication and authorization. Access to all Types (data objects, methods, aggregate services, ML algorithms) should be subject to authorization. Authorization should be dynamic and programmatically settable; for example, authorization to access data or invoke a method might be subject to the user's ability to access specific data rows.

The platform should provide support for external security authorization services. For example, centralized consent management services in financial services and healthcare.



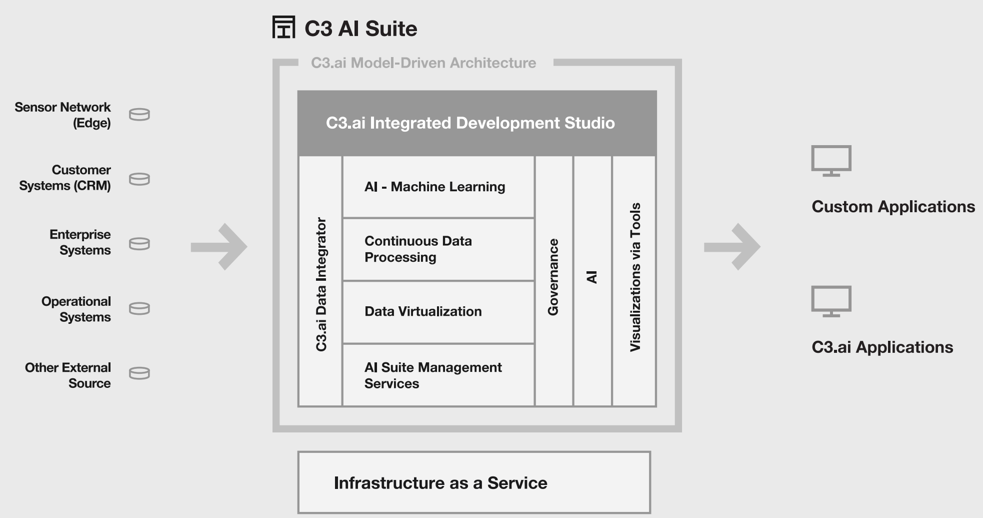
7. Development Tools and Application Management

All Types should be accessible through standard programming languages (R, Java, JavaScript, Python) and IDEs (Eclipse, Azure Developer Tools).

A complete, easy to use set of visual tools are required to rapidly configure applications by extending the metadata repository. Meta data repository APIs are also required to synchronize the object definitions and relationships with external repositories or for introspection of available Types.

Application version control is available through synchronization with common source code repositories such as git.

C3.ai Integrated Development Studio



Next Generation Platform Requirements

Application logic encoded in meta-data

Event points for procedural logic

Integrated Graphical Designers (C3 Tools)

- Visual User Interface Designer
- Data Model Designer
- Object Model Designer
- Integration Designer
- Analytics Designer
- Visual Analytics Designer
- Application Logic Configuration and Scripting
- Data Explorer
- Application Provisioning

Unified, Extensible Type/Object System

- Pre-built Utility Object Model
- Object-Relational/Key-Value Mapping
- Extensible Type System
- Data Model, Object Model, Integration Model, Expressions, Analytic Features, User Interface, App Logic

Database Services

- Relational
- Key-Value Store
- Data Auditing
- Data Versioning
- Pre-built Utility Data Model
- Data Validation

Analytics Services

- Analytic Metrics
- Analytic Expressions/Features
- Predictive Modeling
- Machine Learning Classifiers

Multi-Dimensional Data Services

In-Memory Distributed Data Services

Continuous/Event Processing Services

- Distributed Stream Computing
- Complex Event Processing

MapReduce Service

Application Development Lifecycle Management

- Integrated Graphical Design Environment (C3 Tools)
- Provisioning
- Upgrades
- Monitoring
- Auto-scaling

Integration Services

- ETL
- Synchronous/Asynchronous Messaging
- Integration Process
- REST APIs
- Utility Canonical Object Model
- Canonical Object to Data Model Mapping

Business Process Management

- Workflow
- Orchestration

Business Rules Engine

Private PaaS

- Virtual Private Instance
- Appliance Datacenter Deployment Portal

Mobile, Responsive design

Security Services

- Authentication/Authorization
- Roles/Responsibilities
- Field, Row, Object Visibility
- Data encryption at rest
- Data encryption at rest

Multi-Cloud Deployment

- Private Cloud
- Public Cloud

Security Certifications

- SOC 2 Type 2
- HITrust
- FedRamp
- …

BI Tools

- Tableau
- Qlik
- MicroStrategy

Data Virtualization

- Relational Databases: Oracle, SAP Hana, Postgres, RDS, SQL Server
- Distributed File Systems: HDFS, S3,…
- Application Systems: SAP, Oracle, Maximo
- Key-Value Stores: Cassandra, HBase, DynamoDB, Azure Cosmos DB

Programming Languages

- Java, JavaScript, R, Python, Scala

Data Science

- NLP, Speech, Image, Facial
- Libraries: Scikit learn, Tensor-Flow, Caffe, Torch…
- Supervised, Unsupervised
- Classification, Regression

Data Types

- Structured
- Unstructured: Image, Video, Sound,…

IT Skills for Digital Transformations

With the rapid extinction of corporate data centers and the shift to public clouds providing managed cloud services, concepts such as Serverless and NoOps are rapidly becoming a reality. Staffing IT teams to support a corporate Digital Transformation requires significant re-training and attracting, hiring, and retaining individuals such as data scientists with specialized skill sets.

The table below contrasts the skills of current corporate IT teams and the transformation required to design, develop, and operate next-gen AI applications.

Today

Data Center Management

- Networking, Storage, UNIX, virtualization

RDBMS/Datawarehouse

- Oracle, Sybase, Netezza, Teradata, Exadata

Application Development

- Java, C#, WebSphere, WebLogic, Ruby/Rails, Force, Spring

Enterprise Applications

- SAP, Oracle, Infor

Data Mining

- SAS, SPSS, Excel

Information Security

Digital Platforms

Cloud Platforms (Public/Private/Hybrid)

- AWS, Azure, GCP, Kubernetes, Mesos, Docker
- CPU/GPU/FPGA

Connected Products/IoT

- Gateways, Embedded systems

Relational, Multi-Dimensional and Distributed Data Stores

- HDFS, S3, NoSQL, Columnar, Graph, PostgreSQL, HBase, RedShift, DynamoDB

App Development

- Java, JavaScript, Swift, Android, NodeJS, Angular, React

Enterprise Microservices

- Enterprise Object Modeling and Application Micros-ervices

Data Science

- Classic Machine Learning, Deep Learning, NLP
- Python, R, Spark, TensorFlow, DIGITS

Information Security

Conclusion

The C3 AI Suite is a unique high-productivity, low-code application PaaS for rapidly developing and deploying AI applications at scale across an enterprise. The C3 AI Suite has been developed and hardened through numerous large-scale deployments over 10 years at an investment of $600 million. The C3 AI Suite is proven to support Enterprise Digital Transformations.

The C3 AI Suite's model-driven architecture provides a simplified Type interface, which abstracts underlying technologies through domain specific model representations and expressions and allows application developers and data scientists to collaborate to deliver complex pre-production AI applications in 6 to 12 weeks.

Developing AI Applications requires less code to written and less code to be debugged and maintained, significantly reducing delivery risk and total cost of ownership. Your investment in application code is abstracted from underlying infrastructure and platform services and future-proofed against rapidly evolving software technologies, avoiding lock-in to those technologies.

The C3 AI Suite can be deployed on private clouds and complements public clouds such as AWS and Azure, taking full advantage of their IaaS / PaaS services.

About C3.ai

C3.ai is a leading AI software provider for accelerating digital transformation. C3.ai delivers the C3 AI Suite for developing, deploying, and operating large-scale AI, predictive analytics, and IoT applications in addition to an increasingly broad portfolio of turn-key AI applications. The core of the C3.ai offering is a revolutionary, model-driven AI architecture that dramatically enhances data science and application development.

Proven Results in 8-12 Weeks | **Visit c3.ai/get-started**

C3.ai 1300 Seaport Boulevard, Suite 500, Redwood City, CA 94063