



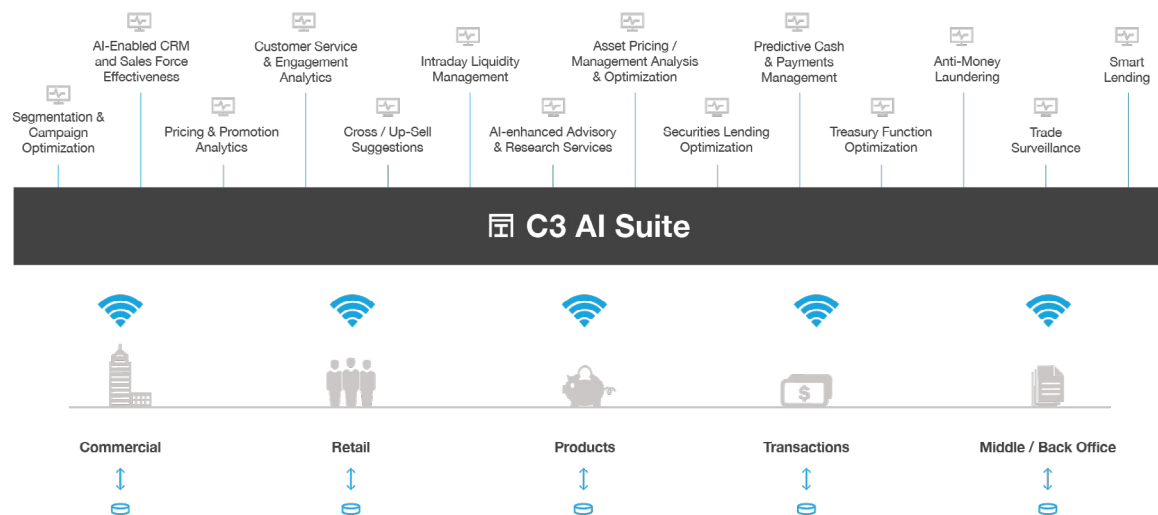
# 10 Core Principles of Enterprise AI

<b>10 Core Principles of Enterprise AI</b>	3
1. Unify All Enterprise and Extraprise Data	4
2. Enable Multi-Cloud Deployments	5
3. Provide Edge Deployment Options	6
4. Access Multi-Format Data In-Place	6
5. Implement Enterprise Object Model	7
6. Enable AI Microservices Across Enterprise	8
7. Provide Enterprise Data Governance & Security	8
8. Support Full Life-Cycle AI Model Development	9
9. Open to 3 <sup>rd</sup> Party IDEs, Tools, and Frameworks	10
10. Enable Collaborative AI Application Development	11

# 10 Core Principles of Enterprise AI

Transformative impact from Enterprise AI comes from applying AI at scale across an organization’s entire value chain. 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. The use cases for AI in banking, for instance, are numerous. 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.

## Representative Value Chain: Banking



The information technology challenges in delivering these AI applications at scale across the enterprise are daunting. Based on C3.ai’s decade of experience helping global organizations apply enterprise AI across multiple industries – including manufacturing, aerospace, oil and gas, defense, healthcare, and utilities – we have identified and codified 10 core capabilities for a complete enterprise AI platform, described below. In a companion document, the [Enterprise AI Buyer’s Guide](#), we have organized the specific requirements for each of the 10 core capabilities into a convenient, comprehensive Enterprise AI Requirements Checklist.

# 1

## Unify All Enterprise and Extraprise Data

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. 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, for example, 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).

To facilitate data integration and correlation across these systems requires a data integration service with a scalable enterprise message bus. The data integration service should provide extensible industry-specific data exchange models, such as HL7 for healthcare, eTOM for telecommunications, CIM for power utilities, PRODML and WITSML for oil and 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, deployments with integrations to 15 to 20 source systems using an enterprise AI platform with a data integration service will typically take three to six months as opposed to years.

# 2

## Enable Multi-Cloud Deployments

Cost effectively processing and persisting large-scale datasets requires 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). An enterprise AI platform must be optimized to take advantage of differentiated services. For example, the platform should enable an application to take advantage of AWS Kinesis when running on AWS and of Azure Streams when running on Azure.

The platform must also support multi-cloud operation. For example, the platform should be able to operate on AWS and invoke Google Translate or speech recognition services and access data stored on a private cloud. It should also be possible for an instance of the platform to be deployed in-country – for example, on Azure Stack – so that it conforms to data sovereignty regulations.

The platform needs to support installation in a customer's virtual private cloud account (e.g., Azure or AWS account) and support deployment in specialized clouds such as AWS GovCloud or C2S with industry- or government-specific security certifications.

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

## Provide Edge Deployment Options

To support low-latency compute requirements or situations where network bandwidth is constrained or intermittent (e.g., aircraft), an enterprise AI platform must enable local processing and the ability to run AI analytics, predictions, and inferences on remote gateways and edge devices.

# 4

## Access Multi-Format Data In-Place

AI applications require a comprehensive set of platform services for processing data in batches, microbatches, real-time streams, 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 enterprise AI platform architecture should allow pluggability of these services without the need to alter application code.

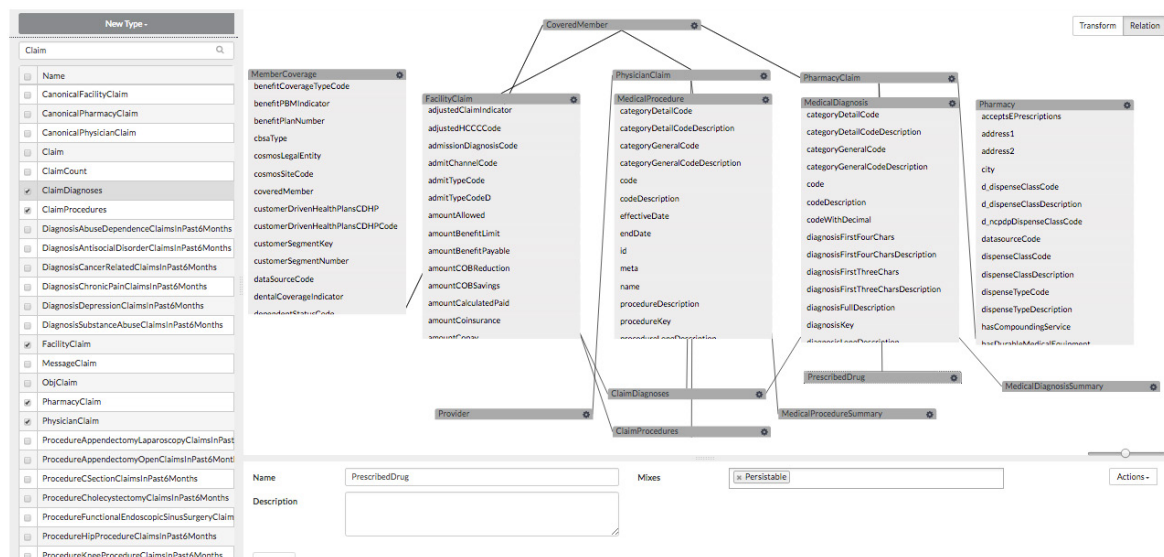
The architecture should also support data virtualization, allowing application developers to manipulate data without knowledge of the underlying data stores. The platform needs to support database technologies including relational data stores, distributed file systems, key-value stores, graph stores as well as legacy applications and systems such as SAP, OSIsoft PI, and SCADA systems.

# 5

## Implement Enterprise Object Model

Re-engineering processes across an organization requires a consistent semantic (object) model across the enterprise. An enterprise AI platform must support a semantic model that 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 must be active and interpreted by the platform at runtime providing significant flexibility to handle object model and schema changes. The platform should support changes to the object model, versioned, and immediately active without need to re-write application code.

### Enterprise Semantic Model



# 6

## **Enable AI Microservices Across Enterprise**

In order to enable developers to rapidly build applications that leverage the best components, an enterprise AI platform must provide a comprehensive catalog of AI-based software services. This catalog of AI microservices should be published and available enterprise-wide, subject to security and authorization access controls.

# 7

## **Provide Enterprise Data Governance & Security**

The enterprise AI platform must provide robust encryption, multi-level user access authentication, and authorization controls. Access to all data objects, methods, aggregate services, and 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 must also provide support for external security authorization services – for example, centralized consent management services in financial services and healthcare.



# 8

## Support Full Life-Cycle AI Model Development

An enterprise AI platform must support an integrated full life-cycle algorithm development experience so that data scientists can rapidly design, develop, test, and deploy machine learning and deep learning algorithms. Data scientists should be able 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 that data scientists can achieve the highest levels of machine learning accuracy (precision and recall).

The platform must enable machine learning algorithms to be deployed in production without the effort, time, and errors introduced by translation to a different programming language. 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.

# 9

## Open to 3<sup>rd</sup> Party IDEs, Tools, and Frameworks

The ability to interoperate with other technologies, products, and components is essential to maximizing developer and data science productivity, enabling collaboration, and keeping pace with ongoing innovation. An enterprise AI platform must be open, providing plug-ins and flexibility for data scientists and developers – including IDEs and tools, programming languages, DevOps capabilities, and others. The platform must support standards-based interfaces (APIs), open source machine learning and deep learning libraries, and third-party data visualization tools. The platform must enable the incorporation of any new open source or proprietary software innovations without adversely affecting the functionality or performance of an organization’s existing applications.

### Open Platform

IDE Support	Programming Language Support	AI / ML Algorithms / Frameworks Support	AI/ML Model Management	Workflow/ Pipelines	APIs	User Interface	DevOps
VS Code ECLIPSE jupyter R JupyterLab	JS python R Java	TensorFlow PyTorch NVIDIA Tesseract OCR Keras spaCy SAS MathWorks	Azure ML rapidminer skyMind mlflow ONNX DataRobot Amazon SageMaker H2O.ai	Azure Analysis Services kafka APACHE STORM spark AWS Kinesis AWS SQS	apigee Azure API Management AWS API Gateway	Power BI alteryx TIBCO Spotfire DOMO +tableau Qlik MicroStrategy React	Azure DevOps elastic Red Hat STIG kubernetes GitHub Jira splunk

# 10

## Enable Collaborative AI Application Development

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 typically conform to IT standards and may require rewriting to a different programming language such as Java. Furthermore, the efficacy of the algorithm is almost always sub-optimal since it has not been tuned against a representative production data set.

To overcome these obstacles, an enterprise AI platform must 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, the platform must enable data scientists to leverage work completed on the platform by data engineers and application developers to handle data cleansing, data normalization, object modeling, and representation. And it must provide microservices to focus on analytic feature development for classic machine learning or deep learning models. The resulting machine learning algorithm should be immediately deployable in production and available as a microservice through a standard RESTful API.

## 10 Core Principles of Enterprise AI

All data objects, methods, aggregate services, and ML algorithms should be accessible through standard programming languages (R, Java, JavaScript, Python) and IDEs (Eclipse, Azure Developer Tools). The enterprise AI platform must provide a complete, easy-to-use set of visual tools to rapidly configure applications by extending the metadata repository. Metadata repository APIs are also required to synchronize the object definitions and relationships with external repositories or for introspection of available data objects, methods, aggregate services, and ML algorithms. Application version control must be available through synchronization with common source code repositories such as git.



1300 Seaport Boulevard, Suite 500, Redwood City, CA 94063