GlobalLogic®
A Hitachi Group Company

Cloud Data Platform Architecture Primer

The foundational concepts that underpin the cloud-based architecture of a modern data platform

Authored by Surbhi Nijhara, Senior Solutions Architect (Technology)

Abstract

This whitepaper focuses on helping you understand and address the building blocks of your data platform journey and how to approach it for architecting. Understanding design considerations can inform your organizational business case and decision-making process

when evaluating the value realization outcomes for a modern data platform goal. This whitepaper provides insight into the architecture that enables you to apply proven methodologies, at-scale benchmarking, cost modeling, and operational efficiency. Thus, realizing the value of modern technology, tools, and workflows that are used for building and operating the data platform.

Contents

Introduction

Building and operating a modern data platform (MDP) is a much-needed prowess for most organizations that o are driving towards digital modernization. The ability to process actionable insights from their ever-growing data and varied sources demonstrates the maturity of an organization in the space of evolving technologies and trends.

Although it is not required for MDPs to always be cloud-based, cloud capabilities often play an essential role in making MDPs for efficient cost models, elastic scalability, and flexible managed services. Typical enterprise data platforms (EDPs) usually exist on-premise, or in hybrid customer data centers made up of traditional data sources like OLTP databases and data warehouses. However, the conventional tools and processes for data acquisition, preparation, and analytical reporting used in EDPs face certain limitations with velocity and variety, hence the integrity of the data.

MDPs created with cloud computing services and cloud-managed data stores provide unlimited object storage, managed relational and NoSQL databases, MPP data warehouses, Spark clusters, Analytics Notebooks, message queues, and middleware. Furthermore, the managed and resilient toolchain and orchestration services from cloud-based platforms enable the process to chain them together seamlessly.

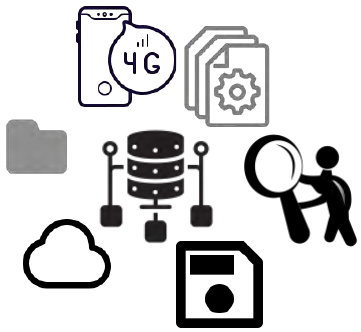
Today, we have decent options from cloud and database vendors who have created promising solutions to let customers process and store huge volumes of data in varied formats using platforms-as-services that adopt a well-architected framework. Customers have less to worry about regarding high availability, scalability, backup, and database operations. However, the platform architecture still needs to align with the specific customer needs, and a suitable contextual blueprint drawn.

This paper aims to take you through the building blocks of a modern data platform architecture and acts as an implementation reference guide for cloud platforms.

Architecture Lens

Architecture considerations require a clear business vision where all the organization stakeholders are aligned to a common goal of providing their customers with a performant and a flexible user experience for everything related to their data.

What should be the focus lens of the architecture team?



Underline Business Outcomes

First, understand and gather the business expectations from the required data platform. The organization may already have a traditional enterprise data analytics platform being used by its customers. Understand the current and potential challenges, the growth rate of data, user experience feedback, competitive index, and last but not least, the cost model of operating the existing platform.

Prototype Early

As you get a comprehensive understanding, start prototyping for early feedback. Avoid wasting efforts on building technology stacks from the grooves. Building solutions from the ground up is expensive, time-consuming, and rarely provides any direct value to your organization.

Ask, Re-Ask and Delegate to Cloud

There are many choices available to opt for cloud-managed services for building and orchestrating the MDP on the cloud. To make the correct choice, some of the below business and technical questions should be answered, at times more than once. This is because the more times you ask, the more answers and details you receive, therefore allowing you to be more decisive on which cloud services to delegate.

- **How many types of data sources connectors are required?** This will help decide if a single cloud-native ingestion service will suffice or other ingestion tools need to be considered.
- **What is the tenancy pattern of the relational data sources and required data sink?** Tenancy Patterns influence the design to a great extent. Single tenancy for analytical purposes is often an ask by the organization and its customers. This can mean a lot of storage and many parallel connections to be established between the source and sink for the initial load, as well as change data capture management.
- **How many customers and tables can be concurrently updated to account for the design of capturing the incremental changes?** This will help choose the appropriate cloud-native workflow services combined with event-based serverless functions versus a single service to orchestrate.
- **What is the acceptable near-real-time and batch window?** This will make you choose the right service configurations and the mechanism to schedule versus trigger.
- **How complex is the transformation logic?** This will help decide if the cloud-provided spark-based APIs can be used or native SQL-based transformation scripts will be more performant and less expensive.
- **Which services are required to exploit the data to realize and deliver value throughout the business?** The answer to this will be useful in deciding the set of analytical tools for exploring your data and unearthing the value of your data.

Measure Continuously

Learn from your prototype and map it back with the business expectations. Without measuring the business's value continuously, you may drift between the possibly changing product requirements that will not justify any expenditure or drive any value from your early testing on the project. However, a number of metrics can be measured to validate the success of the experiments.

An example of this practice was put up for Globallogic's customer Avionte, a leading staffing and recruiting software solution designed for clerical, light industrial, IT, and professional staffing agencies. Avionte, like many enterprise organizations, wanted to improve customer loyalty, improve the success rate of candidate sourcing, produce the best job boards, and onboarding process, apply accurate and compliant payroll, and possess advanced and powerful data insights. This fed directly into their business goal to improve their customer satisfaction index. To achieve this, the team at Avionte identified a requirement to seek feedback from their customers by delivering a Minimum Viable Product (MVP) of their modernized cloud-based data platform.

Some example business outcomes were:

- Number of quality talents per 100 customers per month
- Number of application offers per 1000 customers per month
- Number of payments per 100 customers per month
- Number of payrolls processed per day
- Number of negative customer reviews per week
- Number of custom dashboards and intuitive workflows per customer

Using the feedback loop, updated features, and new service offerings for a better customer experience, we undertook a further design of the modern data platform.

Architecture Blueprint

Once the architectural considerations and measurable metrics are captured, an architectural blueprint should be outlined. This should be followed by prototypes to assess technology, methodologies, and most importantly, exploration of the data to indicate whether the blueprint to deliver the business goals can be incubated into an end-to-end implementation.

There are two primary approaches recommended when building an CDP, each having their own strengths and weaknesses.

The first approach is called a Lambda architecture and has two different components: batch processing and stream processing.

The second approach is called a Kappa architecture where all data in the environment is treated as a stream.

At right is a reference Lambda architecture to fulfill most of the data platform requirements.

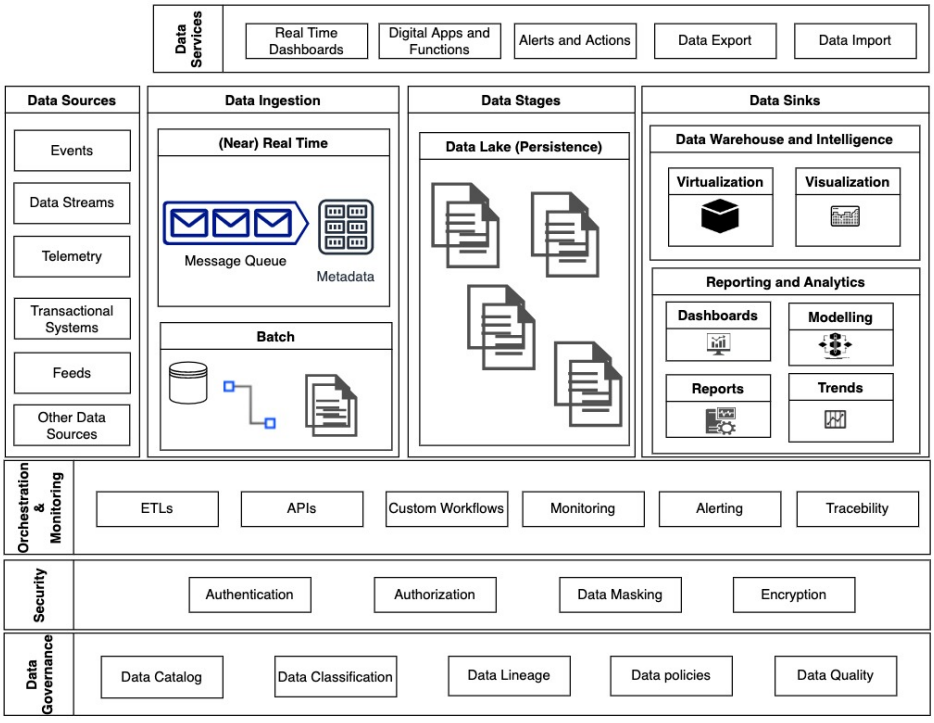


Figure 1 - Data Platform Logical Architecture Blueprint

Because the aim is to get started on building your data platform, let's break down the blueprint into the S-quadruple that are typical in any data platform project:

- **Sources** for Data Ingestion
- **Storages** for Data Processing and Transformation
- **Sinks** for Data Analytics, Visualization, and Machine Learning
- **Services** for Data Access

Defining 4S – Source, Stage, Sink, and Service – the above phases enable us to achieve the functional business layers.

Furthermore, the below layers also span across the 4S to fulfill the well-architected framework for any modern data platform.

- Orchestration
- Security
- Data Ethics and Governance

Let us explore in brief each of the above.

Source

What could be your possible different data sources?

Are they event-based, streams, or 3rd party feeds?

Or are they from transactional systems?

This will help you decide whether real-time streaming or batch-based ingestion mechanism has to be constructed.

Stage

What could the various stages of the data be as it moves from as-is to to-be?

As data moves from data source(s) to the data sink(s), it goes through various stages like raw ingestion, cleansing and deduplication, and transformation. The process is commonly known as ETL or ELT process, as per the context. Each of the stages may require its own staging area for the required processing.

For example, the Batch landing zone shown in the diagram is an intermediate storage area used for data extraction from transactional data sources.

Sink

How do you want to analyze and visualize your data?

What kind of intelligent insights would you like to derive from your data?

Sinks are often data warehouses, data marts, or other data repositories.

Service

Who wants to access and explore the data, and in which form?

Online data services should be throughout the account, especially those requiring raw and processed data from real-time data sources.

Orchestration

How will an ETL pipeline run in an automated way with graceful error handling and logging of important checkpoints?

How will monitoring, alerting, and remediation occur?

They will tackle the needed error recovery and save overall time in the ETL processing through the different stages.

Similarly, security including authentication, authorization, encryption, and data masking should be present across the sources, stages, sinks, and data services.

Security

Who should have access to which environment and what should those access levels entail?

Which regulatory compliances are required to be adhered to?

Which data fields have to be masked for which access?

Data security from various aspects needs to be considered and applied. Secure data is the central pillar of a data platform and no compromise on this front will ever be acceptable by any of the organization's customers.

Data Ethics and Governance

Is your data usable, accessible, and protected?

With time and data growth, is data quality improving?

Is the data management cost decreasing and is data access to data for all stakeholders increasing?

Data governance provides a holistic view of data across five key pillars of observability, including freshness, schema, and lineage.

Reference Cloud Implementation

The data platform architecture can be realized on different cloud platforms entirely natively or using hybrid clouds. In this paper, we will reference the implementation of various data platform components from different cloud vendors such as AWS and Snowflake.

Cloud-based Architecture

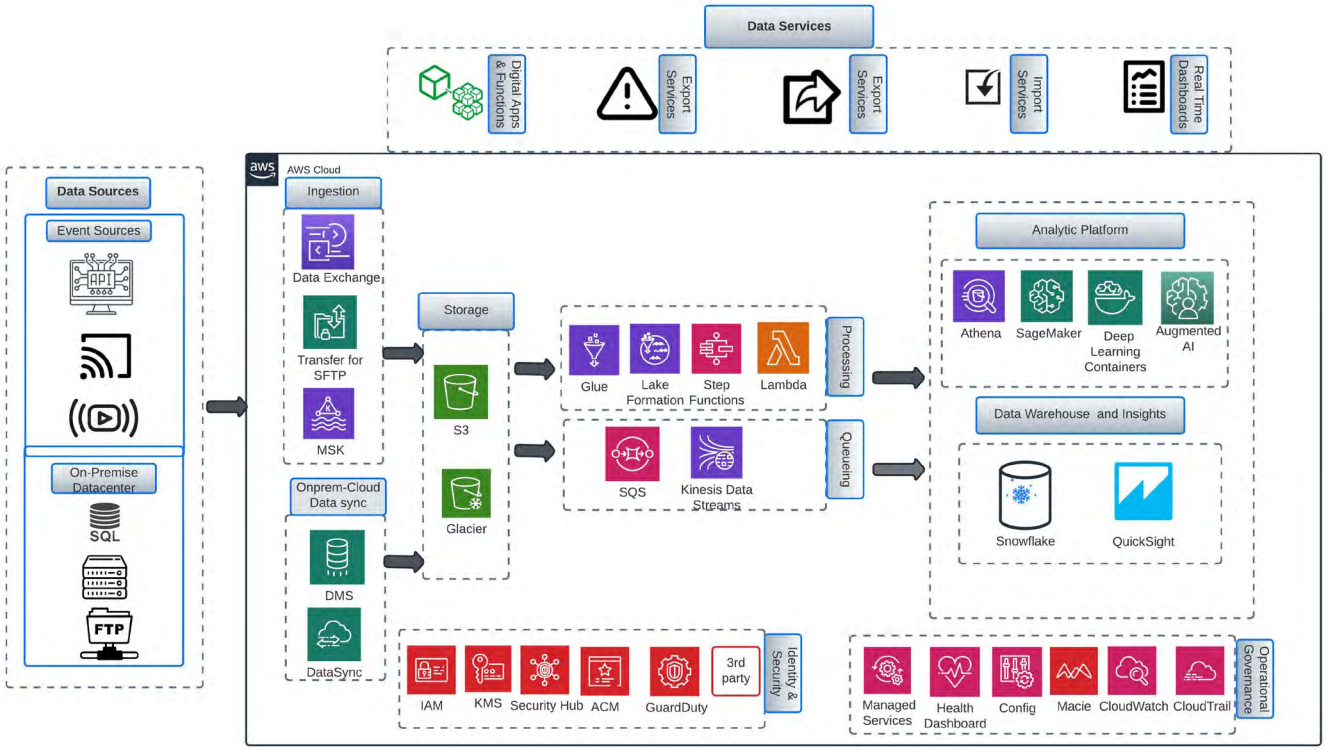


Figure 2 - Reference AWS Cloud Data Platform Architecture

AWS S3-based data like Lake and Snowflake as a data warehouse is used in the reference architecture. For powerful visualizations, Tableau is hosted on an auto-scaling EC2 Cluster. The horizontal spectrum consists of out-of-box AWS offerings for the purpose of orchestration, monitoring, security, and data governance.

Cloud-based Orchestration

As seen earlier in this document, modern data platforms depend on extract, transform, and load (ETL) operations to bulk convert information into usable data. Implementing an ETL orchestration process that is loosely coupled becomes an important design consideration. Orchestration again will depend on your specific sources, stages, and target sinks. See below for a reference implementation using AWS native services.

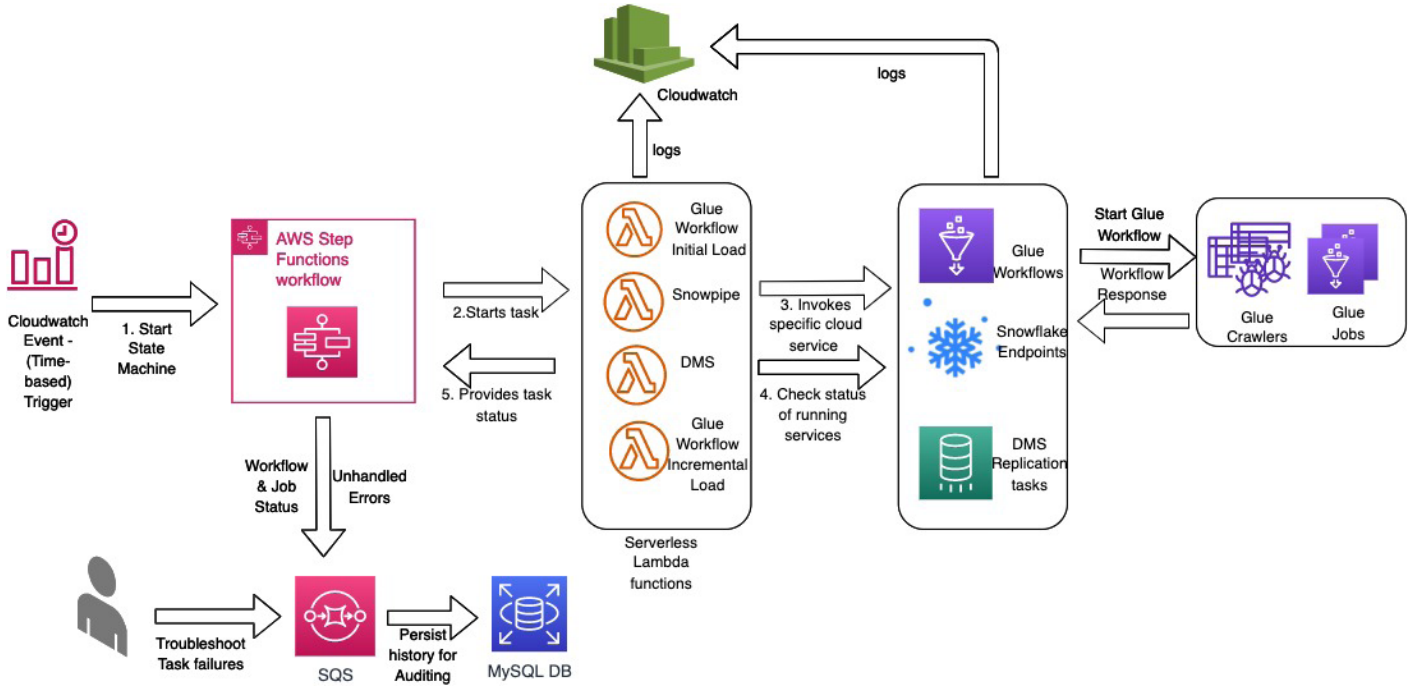


Figure 3 - Reference ETL Orchestration using AWS serverless

Cloud Infrastructure-as-Code

Use serverless computing and Infrastructure-as-Code (IaC) to implement and administer a data platform on the cloud. The following is a reference for the implementation of a Continuous Integration/Continuous Deployment (CI/CD) process throughout the code and infrastructure deployment by using Cloud services such as Azure DevOps and Terraform, a cloud-agnostic IaC tool.

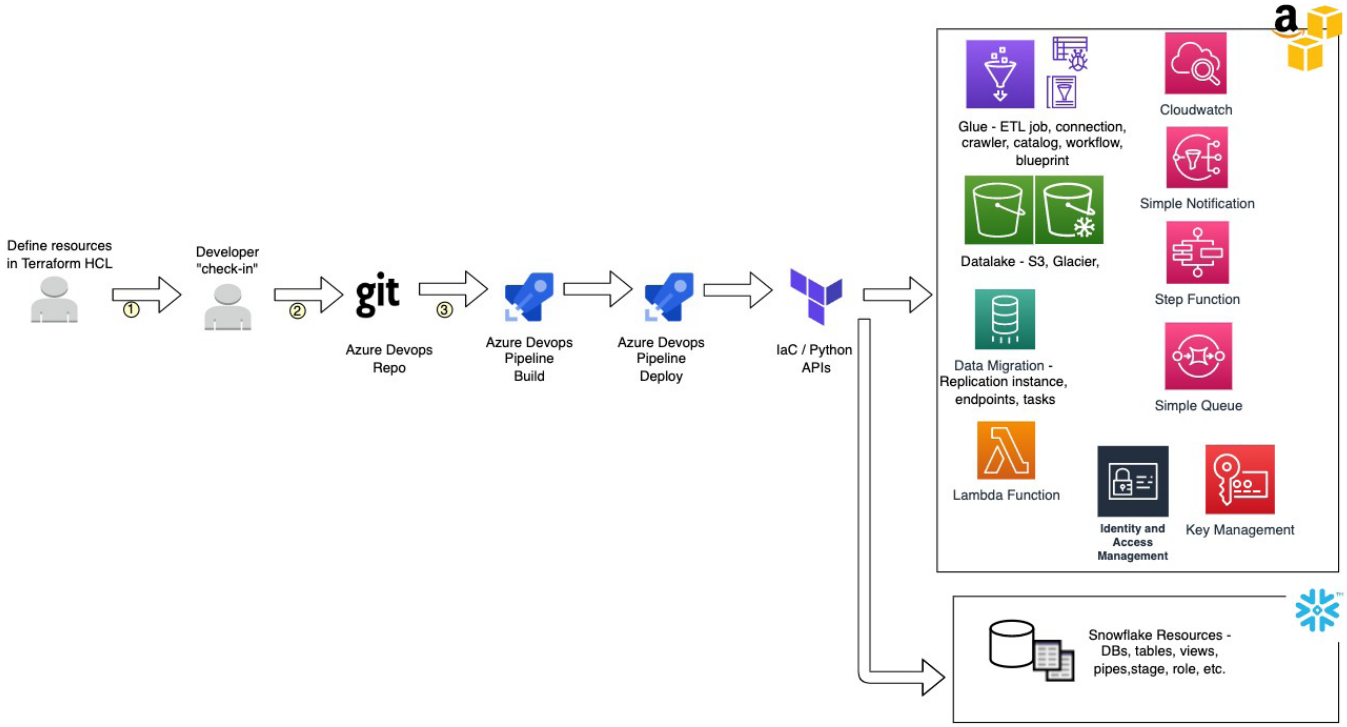


Figure 4 - Reference CI/CD Orchestration for 'Infrastructure' deployment

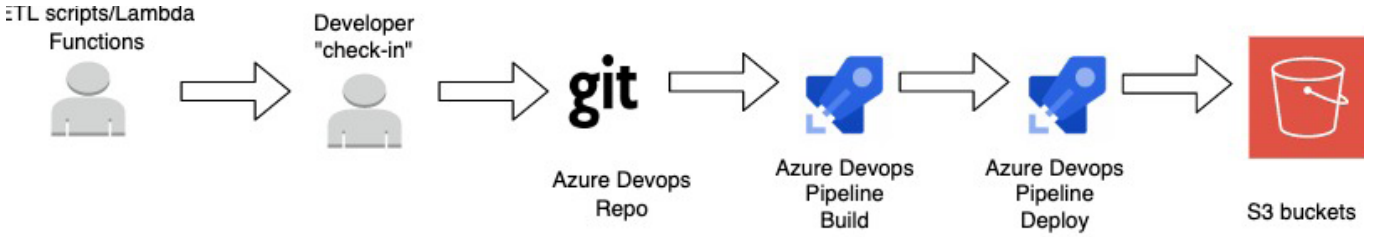


Figure 5 - Reference CI/CD Orchestration for 'Code' deployment

Measure the Designed Pillars

The pillars of building any architecture Includes a modern data platform that stays the same. What differs is the context of the architecture and the knowledge of the frameworks and tools that help to prop up the pillars efficiently and quickly.

Cloud technologies will not only help you to create these pillars but they also ensure it serves your architecture implementation loyally by providing you with various ways to measure and improve them continuously.

Performance and Reliability

- Benchmarking the data pipelines, even during experiments, helps to measure the scalability and reliability of analytics pipelines while predicting the behavior for increased workloads.
- Optimize the runtime of data pipelines with parallel executions for ETL of incremental data.

Operations and Security

- Evaluate the type of storage needs based on access patterns. Create archival and deletion life cycles for data at every stage.
- Monitor the ETL and Analytics pipeline health. Low-to-no error and Exception Index helps to indicate the performance of the platform operations.

Cost Modeling

- Identify the changing workload patterns, velocity, variety, infrastructure usage along with access patterns, and choose cloud-based services accordingly. If these patterns are not initially opted, consider upgrading the architecture technologies to serverless cloud alternatives to reap the benefits of the cost-per-use model.

Conclusion

Customers struggle with starting their platform project because it is difficult to consider design aspects when you have no knowledge, experience, or foresight of their unique requirements as an organization. Without prescriptive guidance, projects fail to get budget approvals, leading to organizations missing the enormous value that data-driven insights can offer.

This whitepaper offers a way forward. We have shown how you can approach the challenge and the unknown. You can use the reference templates to build a comprehensive picture of what your modern data platform will look like. The reference templates can help your organization start a journey towards making data-based decisions during architecture considerations to drive business value, offering benefits for your organization and its customers.

For further reading, please refer to the official AWS documentation, which was used as a resource for this paper.

About the Author

Surbhi Nijhara is a Principal Architect at GlobalLogic, a Hitachi Group Company. She has deep technical skills in Cloud Platform architecture and is a lead consultant in various customer advisories for thinking strategically about cloud solutions to business, product, and technical challenges in enterprise grade digital transformations. She is also co-heads the Cloud and DevOps practice office in technology and architecture for GlobalLogic APAC region.

GlobalLogic®

A Hitachi Group Company

GlobalLogic, a Hitachi Group Company, is a leader in digital product engineering. We help our clients design and build innovative products, platforms, and digital experiences for the modern world. By integrating our strategic design, complex engineering, and vertical industry expertise with Hitachi's Operating Technology and Information Technology capabilities, we help our clients imagine what's possible and accelerate their transition into tomorrow's digital businesses. Headquartered in Silicon Valley, GlobalLogic operates design studios and engineering centers around the world, extending our deep expertise to customers in the automotive, communications, financial services, healthcare & life sciences, media and entertainment, manufacturing, semiconductor, and technology industries.



www.globallogic.com