research

CHECKLIST REPORT

2018

Data Architecture for IoT Communications and Analytics

By David Loshin

Sponsored by:

cloudera®

Transforming Data With Intelligence™

MARCH 2018

TDWI CHECKLIST REPORT

Data Architecture for IoT Communications and Analytics

By David Loshin

TABLE OF CONTENTS

tdwi

Transforming Data With Intelligence™

555 S. Renton Village Place, Ste. 700
Renton, WA 98057-3295

T 425.277.9126
F 425.687.2842
E info@tdwi.org

tdwi.org

FOREWORD

The Internet of Things (IoT) is an architectural paradigm that combines physical devices (with embedded sensors and actuators along with the necessary implementation software), massive network connectivity, data accumulation and analysis, and operational control informed by analytics models that enhance and optimize cross-system interoperation. In turn, a successful IoT deployment can be leveraged to improve the extended system's performance. For example, integrated IoT analytics can help factories reduce or eliminate unscheduled downtime through predictive maintenance, enable manufacturers to improve production quality through continuous monitoring of assembly lines, and anticipate and consequently eliminate power outages across an energy utility's regions by monitoring power usage across a broad network of smart meters. IoT analytics can even alert people to potential health risks by monitoring wearable fitness monitors.

However, systemic objectives for IoT are less about the operation of any single device in the network and more about overall business performance optimization that can be achieved through visibility of all the devices and nodes in the network. The foundation of a successful IoT implementation is a technical architecture that blends network connectivity with an information architecture for streaming, ingesting, filtering, and capturing data. This must be coupled with a means of analyzing the data, creating analytics models, and pushing those models back to the edge nodes in the IoT network. Executing this vision demands organization and discipline when it comes to data management and oversight, requiring information models (including metadata and searchable services) for simplified integration. Finally, the environment must also integrate continuous monitoring to determine when objectives are being met or when there are opportunities for additional improvements.

This checklist explores some fundamental aspects of the data architecture necessary for IoT success. It will examine what is required to enable an environment that can rapidly adapt to the dynamic nature of massive numbers of connected sensors and other end-point devices, communication and data streaming, ingestion and analysis, and deployment of developed analytics models for automated decision making.

Here are eight key suggestions to evolve your data architecture for an IoT environment.

NUMBER ONE

ENVISION, DESIGN, AND GOVERN THE FLOW OF DATA IN MOTION

One of the goals of IoT analytics is to use the data collected from many data streams originating from different individual locations to develop models that can be used to improve *global* business performance across the extended enterprise. The logical end state of the IoT environment integrates many prescriptive models whose actions all incrementally contribute to optimizing overall business outcomes.

Getting to that end state requires a careful examination of the proposed IoT network topology and expectations for how the different connected components interoperate. That includes:

- **Sensors and actuators.** Identify the sensors at the end points of the IoT network, how those sensors are connected, what data streams are generated, and how those data streams are communicated. Identify the actuators located at the end points and the communication methods for controlling them.
- **Edge nodes.** In many IoT architectures, localized computing resources are used to accumulate sensor data streams and package them together to be forwarded to a centralized computing and analytics system. In addition, edge nodes provide a site for localized streaming analytics models for oversight and control of the end nodes within an assigned jurisdiction.
- **Centralized server(s).** These are the main systems for accumulating and analyzing streaming data from across the entire network and creating the models used to manage automated decision making.

Devise a data flow architecture that models the network connectivity—how the information from the different streams is accumulated and managed, how the various data streams are connected through edge nodes to a centralized platform, and how analytics models are created and subsequently integrated back into the network. You must also account for the realities of bandwidth dynamics, especially designing data flows in an environment where the sensors are not necessarily static, but moving.

In turn, ensure that your selected platform architecture supports the end-to-end data flow aspects of IoT data communication: routing and streaming data from the network end points to the centralized analytics platform; introducing bandwidth controls (such as back pressure controls and dynamic data package reduction when bandwidth is low or throttling packets until the bandwidth improves); data acquisition, transformation, filtering, and analysis; and deployment of analytics models. Choose a platform that facilitates the data flow across different types of devices and

machines, captures data source and transformation provenance (to support traceability and consistency management), and allows you to control and manage the end-to-end data life cycle across the environment.

NUMBER TWO

LEVERAGE INTELLIGENT STREAM PROCESSING

Any IoT environment is going to involve many types of devices, embedded sensors, and computing nodes, all sharing one feature in common—the continuous generation and communication of data. Analyzing data ingested from multiple simultaneous data streams not only allows you to develop a holistic perspective of what is happening across the environment, it also contributes to a richer analysis used to improve the precision of analytics models to be directly embedded within the network.

There are two aspects of stream processing that are integral to the success of an IoT application. First, an IoT analytics application must be able to handle the ingestion and capture of many streams of fast-moving data. Second, the resulting analytics models must be deployed across multiple tiers in the IoT network to continuously monitor the data streams, supporting automated decision making and generating notifications that drive actions to improve the business.

Data streams in an IoT network are more than just sources of information conveyed to a centralized server—they fuel the continuous analyses at different locations across the grid of devices. Yet there is fluidity among the participants in an IoT network—new devices join while others are disconnected, along with intermittent and unexpected changes in interface specifications. Therefore, data stream handling requires a greater level of sophistication and intelligence to ensure coherence, synchronization, validation, and integration. Above and beyond the mechanics of data stream ingestion, consider these aspects of intelligent stream processing:

- **Profiling.** This includes data value frequency analysis and interpolation of formats and structure that can be validated against documentation (if any exists)
- **Semantic metadata.** Augment the structural metadata with data element definitions and contexts
- **Curation.** Data curation is the process of assembling, organizing, managing, and ensuring the usability of a collection of data streams

Intelligent stream processing blends technologies for streaming (such as Apache Kafka or Storm), developing and mapping data flows (such as Apache NiFi), and scalable parallel processing (such as Spark) with data discovery and profiling tools.

NUMBER THREE

DEVELOP A METADATA AND SERVICES CATALOG

The potential diversity of the types of devices and computing engines in an IoT network is often muddied by the absence of clear documentation about the structures and data formats embedded within each data stream, let alone the semantics associated with the information contained within those streams. Our second item suggested the need for "intelligent" stream processing for consistent and accurate data source integration. This intelligence depends on a vast inventory of metadata associated with each of the data sources streaming data across the IoT network, coupled with the details of how data policies and rules are to be integrated within the data flow. In essence, developing and implementing an intelligent system requires aspects of data governance.

Data governance is a set of core principles, processes, and tools that ensure that rules and policies related to data management are applied consistently across the different levels of the IoT network. Pragmatically, IoT operational data governance can be actualized using a shared, searchable catalog that captures data policies, application programming interfaces (APIs), structures and formats, business rules, and semantic metadata for each device's data stream. Centralizing this metadata and data services catalog not only enables applications to automatically discover the APIs and data stream schemas, it effectively simplifies the ability to access, ingest, and analyze incoming data. As new services are developed, they can be registered within the catalog and made available for other data consumers.

This may sound simple, but in practice it requires discipline to document this information and keep it up to date, especially due to the dynamic quality of the myriad of data streams. Many streams will ultimately converge at a centralized location for analysis, but it would be difficult to constantly manually inspect the data to ensure its validity, especially if transformations have been applied at different stages within the network. By capturing data lineage and provenance in your metadata and services catalog, you have the details about the data flows so that emergent anomalies at the centralized location can be quickly traced to their sources for assessment and remediation.

NUMBER FOUR

ENABLE VERSION MANAGEMENT TO SUPPORT AGILE DEVELOPMENT AND DEPLOYMENT

Although the nodes and devices in the IoT environment are all constantly generating data, that is where their similarities end. Not only is it unlikely that different types of devices will conform to the same standard for data representation, each device's data stream may be modified as the device is enhanced or updated. From the perspective of the central points in the IoT environment, navigating the difficulties in this rapidly changing landscape is challenging.

Many data streaming applications make the mistake of hard coding data stream schemas and the ways information is packaged and unbundled at different phases of the data flow. Hard coding the interface management prevents reusing the embedded schema, complicates system maintenance (by making it difficult to figure out where changes need to be made), decreases overall system development flexibility, and is resistant to the kinds of necessary governance previously described.

Address these challenges by versioning the schemas of the incoming data streams within the metadata and services registry. Benefits to enabling discoverable interfaces for different versions of the many data streams include:

- Allowing ingestion of data streams from concurrently operating similar devices running different versions of their operating systems
- Supporting agile development in rapidly adapting to unannounced changes in the environment
- Exposing the differences between versions to allow application developers to evolve their systems at different rates while remaining consistent across the IoT network
- Easier implementation of ingestion of data streams from new devices that are rolled into the network

Note that versioning of schemas is not limited to the structures and formats of data within the messages conveyed within each data stream. The concept of the versioned schema can be extended to include implementation templates (mapped to the different system components such as NiFi flows or Kafka streams), declarative business rules that might change in relation to changed structures or formats, or even different versions of analytics models that are tuned to different stream versions.

NUMBER FIVE

SIMPLIFY AND SECURE DATA INGESTION

The scale of ingested IoT data is massive, requiring a big data analytics environment leveraging components in the Apache Hadoop ecosystem (such as the Hadoop programming environment, YARN, and Apache Spark), distributed file systems, components for ingesting static and dynamic streaming data, data transformation tools, and analytics libraries. To take best advantage of these big data environments, it is critical to simplify the processes by which data sets and data streams are ingested, filtered (if necessary), persisted, and forwarded to downstream algorithms and tools to create those event-processing models for streaming data that will be deployed across the network.

It is critical to enforce data protection controls by implementing a variety of data policies for data asset classification associated with levels of sensitivity (such as protected personal information or confidential intellectual property data), data protection policies (such as encryption at source or data masking), and role-based access controls that are bound to the data classifications and policies. The scale and breadth of IoT—including the need to manage and automate a massive number of bidirectional, point-to-point data flows—makes extending the data protection perimeter across the massive number of nodes particularly complex. In addition, realize that the sensors and devices in the network may operate in imperfect environments, leading to sensor data that is incomplete or flawed.

Data governance practices such as monitoring provenance are critical to ensure that faulty transmissions can be restarted when necessary to ensure data integrity across all stages in the network. Technologies such as Apache Ranger, Atlas, and NiFi let you orchestrate, manage, and validate data as well as secure the data passageways from the edges through to the center. These tools let you push the security and protection to all the nodes in your network and, in the case of MiNiFi, to the edges and end points (i.e., the devices themselves), which are limited to a much smaller footprint due to the devices' size and resource constraints.

NUMBER SIX

EMPLOY ADAPTIVE ANALYTICS MODELING

One beautiful aspect of an IoT system is how it unites a variety of "always-on" devices and integrates intelligent, continuously monitoring event-processing models that generate appropriate actions to optimize outcomes. Yet although developing analytics models that monitor continuous data streams is complex enough, typical approaches sometimes neglect one of the most critical (yet obvious) aspects of the environment: managing state.

The state of an IoT system comprises the persistent information about the collection of connected device states, especially in terms of internal variables or reported environmental measurements (such as a device's temperature, ambient air quality, humidity, etc.). Analytics models take their relevance from the state—the streams are monitored for changes in the state that indicate necessary actions.

Models must account for both the immediate snapshot of the streaming data as well as how the state of the environment has changed within a defined recent time frame. For example, detecting a rise in temperature along one manufacturing assembly line might generate an alert to check that line's status, but simultaneous temperature increases across multiple lines in the same building over a five-minute span might generate an alarm that the building is on fire.

These real-time analytics environments track time-series data and maintain history, and adapt according to two different duration windows. Streamed messages within a *tumbling window* are blocked by a time frame, such as all device readings within a 10-second interval. Any message will be assigned to a single tumbling window. In a *sliding window*, the time frame of a specified duration slides across the sequence of messages. A sliding window of 15 seconds with a sliding interval every three seconds allows sets of messages to be evaluated every three seconds but also allows messages to belong to more than one sliding window.

Employ adaptive analytics modeling by incorporating techniques allowing models to review streams along sliding windows. Look for technologies that cache or stage streamed data in memory yet do not introduce artificial delays into the data flow.

NUMBER SEVEN

PUSH STREAMING ANALYTICS MODELS TO THE EDGE

A naïve vision of an IoT environment presumes a massive field of devices streaming data to a centralized platform that ingests and analyzes data to create analytics models. This single centralized system would operate thousands (or more) of these models, monitoring a complex combination of simultaneous continuous data streams.

In reality, centralizing model deployment and automated decision making is unwarranted for a number of reasons, including:

- **Complexity.** Different types of analytics models expect inputs from a variety of data streams. Models tuned to enterprisewide decisions may process many streams while models overseeing the operation of a specific device at a specific location only monitor that device's input stream. Trying to coordinate the connection of all these streams is complex.
- **System performance.** The central system must be properly sized to meet the real-time performance needs of hundreds or thousands of simultaneously executing models.
- **Cost.** Accordingly, the more resources the system needs, the more costly it will be.
- **Slowed time to decision.** Even with the fastest central system, you still need to account for the latency associated with moving data across the network. The farther the model is from the device, the longer it will take to communicate the decision back to the device.

Realize that the data from each of the devices must stream through the computing nodes that form the edge of the IoT network on the way to the central destination, and in many cases the models used to monitor and regulate the devices in an environment are best deployed in proximity to those devices. For example, although models for monitoring energy consumption might be centrally created, based on aggregated data streaming from all the devices in a collection of smart buildings, decisions about device power regulation in each building really only depend on the data streams generated by the devices in that specific building.

Because the edge nodes are much closer to the collection of devices they govern, pushing your analytics models to these edge computers reduces implementation complexity, improves system performance, and, most important, speeds the time for taking actions.

NUMBER EIGHT

EMBED CONTINUOUS BUSINESS PERFORMANCE MONITORING

The goal of deploying analytics models at many locations across the IoT network is to analyze data streams and generate alerts about emerging risks (such as imminent part failure) or identify emerging opportunities for improvement (such as increasing production at particular factories to meet growing product demand). However, how can you tell if your models are working to achieve the desired business objectives?

Clearly, because the analytics models are directly integrated within nodes of the network, the IoT computing paradigm is amenable to other types of embedded process instrumentation. That gives you the option of augmenting the environment with processes to log automated decisions resulting from analytics models. These logs can be continuously evaluated to ensure that the developed and deployed models are producing the anticipated results.

Interestingly, logging decision events and transmitting the logged information to the centralized server introduces additional data streams into the IoT network! Yet these new data streams play an important role in enabling a feedback channel that informs the analysis engines and allows data scientists to tweak the algorithms and help produce improved models that can be published back to the nodes within the network. Of course, as suggested in Number Four, these models can be versioned and implemented in different locations so that their outcomes can be compared. This helps further refine the optimizations and performance of IoT analytics.

AFTERWORD

This checklist has reviewed a number of key suggestions for evolving the data architecture for an IoT environment. Visualize the intent of the IoT network and use tools that help continuously refine the network's topology and map out the data flows. Develop a metadata and services catalog that logs information to simplify device integration. These ideas, as well as versioning of data schemas and services, together support an agile approach to determine where and how to process the data streams and rapidly support ingesting data streaming from and back to the dynamic collection of devices. Institute data governance practices that allow consistent enforcement of data policies for data quality, timeliness, security, and protection at all points in the network—from the end points through the edge nodes, all the way to the centralized system.

We recommend that IoT data architecture embrace the types of technologies that support these suggestions, specifically a data services catalog, data source integration, integrated security controls, data governance/stewardship, and data life cycle management. This will support the ingestion, processing, and analysis of massive numbers of data streams, resulting in analytics models that guide profitable actions with automated decision making. Incorporating continuous performance monitoring will allow your analysts to improve their machine learning algorithms and provide a positive feedback loop to refine and improve those models.

ABOUT OUR SPONSOR

cloudera®

cloudera.com

At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises. Learn more at cloudera.com.

ABOUT THE AUTHOR

David Loshin, president of Knowledge Integrity, Inc. (www.knowledge-integrity.com), is a recognized thought leader, TDWI instructor, and expert consultant in the areas of data management and business intelligence. David is a prolific author regarding business intelligence best practices; he has written numerous books and papers on data management, including *Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph* and *The Practitioner's Guide to Data Quality Improvement*, with additional content provided at www.dataqualitybook.com. David is a frequent invited speaker at conferences, online seminars, and sponsored websites and channels including TechTarget and The Bloor Group. His best-selling book, *Master Data Management*, has been endorsed by many data management industry leaders.

David can be reached at loshin@knowledge-integrity.com.

ABOUT TDWI RESEARCH

TDWI Research provides research and advice for BI professionals worldwide. TDWI Research focuses exclusively on analytics and data management issues and teams up with industry practitioners to deliver both broad and deep understanding of the business and technical issues surrounding the deployment of business intelligence and data management solutions. TDWI Research offers reports, commentary, and inquiry services via a worldwide membership program and provides custom research, benchmarking, and strategic planning services to user and vendor organizations.

ABOUT TDWI CHECKLIST REPORTS

TDWI Checklist Reports provide an overview of success factors for a specific project in business intelligence, data warehousing, analytics, or a related data management discipline. Companies may use this overview to get organized before beginning a project or to identify goals and areas of improvement for current projects.