## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>4</td>
</tr>
<tr>
<td>ORCHESTR8 PLATFORM</td>
<td>6</td>
</tr>
<tr>
<td>SYSTEM ORGANIZATION</td>
<td>7</td>
</tr>
<tr>
<td>THE HARD PROBLEMS</td>
<td>8</td>
</tr>
<tr>
<td>ORCHESTR8 ARCHITECTURE</td>
<td>10</td>
</tr>
<tr>
<td>THE ADMINISTRATION DOMAIN</td>
<td>12</td>
</tr>
<tr>
<td>THE SERVICES DOMAIN</td>
<td>14</td>
</tr>
<tr>
<td>THE EXPERIENCE DOMAIN</td>
<td>25</td>
</tr>
<tr>
<td>ORCHESTR8 DEVOPS &amp; OPERATIONS</td>
<td>29</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>33</td>
</tr>
</tbody>
</table>
Orchestr8

INTRODUCTION

THE LAST FORTY YEARS HAVE DELIVERED TREMENDOUS ADVANCES IN COMPUTER PROCESSING POWER, STORAGE CAPACITY, AND DISPLAY TECHNOLOGY.

While these advances have solved increasingly complicated problems, the solutions have generally required increasingly complicated user interfaces to operate them.

The software industry has responded by developing new interface techniques and new interaction hardware to enable more appealing and usable human interfaces. Academically, computer scientists, cognitive psychologists, and others involved in human-computer interaction have shifted from focusing on user interface design to more holistically understanding user/customer experiences. Naturally, technology companies have responded by offering products to help with the design and delivery of improved experiences.

Level 11 has been in the customer experience business since before the term was popularized. We have been heavily involved in several large projects aimed at transforming hospitality and retail businesses by providing fundamentally better experiences. Most notably, we are one of three companies recognized as key architectural contributors to the Carnival Ocean Medallion project. The Level 11 Orchestr8 platform is the product of our learning from these projects—the generalized concepts that apply to a broader set of problems. Many parts of the platform are components that have been independently licensed to existing clients over the course of the last half-decade, while other components have been purpose-developed as integrated components of our unified platform.

In addition to our expertise with experiences and user interface design, we have also developed a strong understanding of the importance of emotion and narrative. We recognize that customers aren’t just using interfaces; they are performing actions within the context of a larger goal, and these actions have an inherent narrative to them that needs to be recognized and incorporated in all of their interactions.

Within these narratives, there is an active relationship being curated between the customer and the sponsoring entity. Appropriately framed, it is a relationship that sets the correct tone for interactions and services. A commercial entity, if it is to thrive in today’s marketplace, must embrace its role as the Host for the duration of its engagement with its customers. The human within a Host’s venue is then naturally the Guest—honored and invited. The Host’s staff, empowered by the digital experience, continue to be the tangible champions of the Guest-Host relationship and represent a key user group within high-value experience ecosystems. Establishing this Guest-Host dynamic is foundational for determining the lengths that a Host will go to in order to design and achieve the sense of confidence, helpfulness, and grace that causes Guests to have genuinely great experiences.

As such, Orchestr8 is more than a user interface technology or a location tracking system (although it includes both of these). The most important aspect of the platform is the notion of Guest context. All the components that interact to deliver positive Guest-Host interactions have deep knowledge about the Guest. This context includes more than just the Guest’s profile;

---

1 Carnival Ocean Medallion project
   https://www.princess.com/ships-and-experience/ocean-medallion/

2 DESIGN-DRIVEN COMPANIES OUTPERFORM S&P BY 228% OVER TEN YEARS - THE ‘DMI DESIGN VALUE INDEX’
   http://www.dmi.org/blogpost/1093220/182956/Design-Driven-Companies-Outperform-S-P-by-228-Over-Ten-Years-The-DMI-Design-Value-Index
it includes information about the Guest’s journey and genome. Knowledge, for example, that the Guest has an appointment in 30 minutes or that the Guest is attending a conference, provides the Host with temporal clues to the Guest’s constraints and motivations. The Guest genome is our term for all of the information we have concerning the Guest’s preferences and likely behavior. This information is frequently provided by machine learning algorithms that develop models of behavior used to predict Guest actions or to evaluate Guest reactions.

The Level 11 Orchestr8 platform can operate as the foundational basis for any customer experience solution. This document provides a more detailed description of the platform, introduces some of its most fundamental concepts, and provides an example of how it can assist in the delivery of a rich and satisfying customer experience.
THE ORCHESTR8 PLATFORM

Orchestr8 (O8) is architected to be a loosely coupled “system of systems” designed to be highly available, evolvable, scalable, and secure.

An O8 installation is a single deployed instance of O8 which typically manages the location services and other digital capabilities for a single physical location.

O8 is a modern distributed computing architecture that follows modern DevOps and microservices best practices. The O8 engineering philosophy can be summarized as:

- All system components should be integrated in a manner enabling their easy augmentation or replacement
- To the greatest extent possible, solutions to common hard problems should be standardized by the platform
- Best-of-breed open source systems provide the core technology stack
- Integration points for valuable paid software and online services are provided.
- Incorporation of DevOps best practices into the O8 management plane is key to reliable and scalable deployments
- O8 should enable subsystems to be deployed into any modern cloud or on-prem environments in any arrangement desired
SYSTEM ORGANIZATION

O8 is comprised of three high-level subsystems known as **domains**. Each domain has numerous additional subsystems and components of its own specific to its role in the larger system:

- **Administration Domain** - management plane for an O8 installation
- **Services Domain** - backend services supporting capabilities of the the Experience and Administration Domains
- **Experience Domain** - mobile and web end-user facing components in service of both Guests and the Host’s Staff

As a platform, O8 is designed to enable highly interchangeable component subsystems to work seamlessly and reliably with each other. This loose coupling permits the system to be simultaneously both extensible and evolvable.

At a high level, adding new features to O8 involves creating, deploying, and unlocking **capability modules**. Each module contains the necessary configuration and code needed to support the addition of a complete feature set with functionality and resources deployed dynamically into potentially all of the system domains.

Key to this approach is the standardization of module interfaces with regards to presentation layer plugins (UI), code deployment, system networking, service discovery, and communications methods. This approach to standardization extends to critical system aspects that crosscut modules such as security, localization, privacy, and disaster recovery.
THE HARD PROBLEMS

ALTHOUGH BUILDING MODERN SOFTWARE SYSTEMS IS CHALLENGING IN MANY WAYS, THERE ARE FOUR HARD PROBLEMS THAT O8 ADDRESSES IN A SYSTEMATIC FASHION THROUGH ADOPTION OF APPROPRIATE TECHNOLOGIES AND TECHNIQUES.

| ASYNCHRONOUS INTERACTIONS | CONCURRENT INTERACTIONS | ERRORS | SYSTEM STATE |

ASYNCHRONOUS INTERACTIONS

Complex system behavior can be categorized in many ways, but a fundamental consideration is whether interactions between components are synchronous or asynchronous. In a synchronous interaction, a component that starts communicating with another component waits for a response before continuing to do other work. In contrast, asynchronous interactions between components don’t wait for a response but expect one later. The challenge with asynchronous interactions is that the response can come at any time in the future—or possibly never at all. Therefore there is no knowing what the caller will be doing when the response arrives, which makes designing these types of interactions challenging.

While it may sound like synchronous interactions are obviously better because they are simpler, they are also more fragile in that the components

1. Are intimately tied together during the interactions
2. Do not effectively utilize computing resources.

Therefore, tackling the challenges of asynchronous interactions at the platform level is a key value proposition of O8. To accomplish this goal, O8 is designed to assist component developers implement patterns of behavior based on standardized communication methods, enterprise messaging, state management protocols, and logging mechanisms. These patterns and capabilities are implemented or utilized by all O8 services and components, resulting in increased consistency of module implementations and systemic reliability in production.

CONCURRENT INTERACTIONS

While asynchronous interactions are hard, interacting with multiple other components at the same time is even harder. The challenge with asynchronous interactions is time—when the response will come is unknown. The challenge with concurrency is order—which response will come next is unknown.

Once again, the choices made in the technologies and their utilization in the O8 platform assist module developers (both internal and third-party) to think about these problems in a disciplined and holistic fashion.

The O8 architecture is inspired by an advanced software systems architectural principle known as the Actor model. The essence of this approach is the idea of independent components communicating through messages. Interestingly, the Actor model helps not only with properly managing concurrent interactions but, by its use of sending messages for component interactions, embraces asynchronous communication between components as an axiomatic attribute of the architecture as well.

The Actor model is highly compatible with our focus on flows and transaction managers.
These concepts are similar, or from some perspectives identical, and are defined by the existence of an entity in the system that is dedicated to managing a multi-staged set of work that potentially results in multiple different outcomes. The Actor model (message passing to independent actors) combined with flows (actors that manage work processes) is a key aspect of the O8 architectural robustness and scalability.

**ERRORS**

Despite all best efforts, stuff happens. Even if a system is initially perfect (never the case), over time changes occur in complex technology ecosystems and the unexpected will come up.

To address the inevitable unexpected situations, O8 has state-of-the-art monitoring and alerting technologies integrated deeply with each system module. Errors are captured for support evaluation and resolution. Combined with sophisticated logging, alerting, and visualization technologies, O8 enables support technicians and developers to quickly understand the context in which the error happened. This enables quick resolution of errors either through manual intervention or the updating of system modules.

**SYSTEM STATE**

In computer science the term *state* is used in a number of similar but not identical senses. In all its meanings, however, state management is at the heart of all the hard problems discussed in this paper as well as their solutions.

Best practices dictate that modern scalable and highly available systems be comprised of individual services and components that are designed to be stateless—that is they do not keep important data in memory that, if lost due to a crash, would cause systemic problems going forward. This approach is central to the O8 architectural philosophy for module design.

Despite the processing components being stateless, the overall system does have state that must be preserved in order to remember what happened in the past and to determine what needs to happen in the future. This requirement is partially met by writing data to durable storage systems after computation is complete. The reliable persistence of raw data can be a relatively simple matter of proper configuration of redundant storage. Highly durable storage can be straightforward to set up for a given deployment configuration. Regardless of the ease of configuration, however, durable storage for big data still requires planning and ongoing monitoring to ensure it is not exceeding budgets or underperforming other metrics.

What is also challenging to design are flows of activity that must properly capture critical business logic. These flows are another sense of the term *state* and can be a harder problem to solve than reliable data storage. What makes flow state very complex is properly accounting for the aforementioned other hard problems in the flow—asynchronous and concurrent interactions matrixed against expected failures and unexpected errors. When those challenges entangle multiple distributed systems, designing flows to successfully handle all possible outcomes is the hallmark of quality in software system architecture.

O8 uses a number of modeling methodologies to design and implement flow state. Critical business flows are designed using disciplined software engineering practices including BPMN 2.0 and UML flow models. These techniques permit the validation of flow states with the client before full implementation in code. Additionally, they also assist in system analysis, debugging, and issue resolution of both test and production system behavior.
Now that we have explored the big picture of the Orchestr8 platform including broad goals, general approaches, and first principles, a deeper dive into the architecture is in order.

Orchestr8 has three high-level domains:
Below we can see a diagram showing the relationships between the domains and the component subsystems comprising each domain:

Orchestr8 – ARCHITECTURE

Orchestr8 Experience Platform Architecture

ADMINISTRATION

EXPERIENCE

SERVICES

CONTEXT LAYER

INFRASTRUCTURE LAYER

INTEGRATION

User Touchpoints:
- Apps
- Mobile
- Web
- Kiosk
- Wearables
- NFC Bands
- Etc.

API Gateway / Orchestration Layer / Security

Experience Management:
- Feature Unlocking
- Content Management
- Content Display
- White Labeling

Core:
- Profile
- Dossier
- Groups
- Folio
- Payments
- Social Platforms
- Calendar

Domain Expertise:
- Location Events
- Environmental
- Motion Analysis
- Time Sequence
- Wayfinding
- Map Services
- Feedback & Ratings
- Photo/Video
- Loyalty

Integrated Services:
- Business Intelligence
- Machine Learning
- BPMN Workflows
- Analytics
- Vendors
- Promotions
- BMS
- Hospitality
- Transportation
- ERP
- Parking
- Security
- CBM
- Recommendation Engine
- Guest Genome
- Events
- Rules

Experience Intelligence Infrastructure
- Cameras
- Motion
- Environmental
- Heat Sensors
- External Events
- Websocket
- TCP/UDP
- REST/HTTP
- Enterprise Messaging

Interprocess Communications

Data Storage
- Relational
- Document
- Geospatial
- Media
- Data Vault

Networking Infrastructure
- Wired
- Bluetooth/BLE
- NFC
- Wi-Fi
- RFID
- Cellular

Archival Backup / Cold Storage

Orchestr8 Experience Platform Architecture
THE ADMINISTRATION DOMAIN

THE ADMINISTRATION DOMAIN ENCOMPASSES ALL ASPECTS OF OPERATIONAL MANAGEMENT OF AN O8 INSTALLATION.

Core administrative capabilities exposed here include:

- ROLE-BASED USER MANAGEMENT
- FEATURE UNLOCKING FOR THE MANAGEMENT AND CONFIGURATION OF MODULES
- PERFORMANCE MONITORING
- ANALYTICS
- IOT DEVICE MANAGEMENT
USER MANAGEMENT

O8 supports multiple technologies for managing users via flexible and efficient access control policies, including role-based access control, attribute-based access control, and on-behalf-of privileges. Leveraging integrations with existing HR and access management systems allows for automated real-time access management to avoid any lingering access rights and security holes and automating integration with partners.

O8 delivers a seamless, personalized, and secure user experience for Guests by providing a single view of their identity.

FEATURE UNLOCKING

Feature unlocking is a powerful component of the platform that gives administrators control of the use of a feature in a fine-grained way. Administrators can decide which module features are most appropriate for their installations and add or remove them from their deployments at will. Modifying these settings will make these features appear and disappear from both the web and mobile experiences.

Unlocking a feature in Orchestr8 enables the new functionality across the Services and Experience Domains. At that time, the Administration Domain may also display additional management tools controlling access, management and configuration of the new capabilities for the unlocked feature.

PERFORMANCE MONITORING

An O8 installation provides a Host's administrators with the ability to monitor, troubleshoot, and diagnose performance issues in their applications. Alerts can be configured and integrated with your existing team applications. O8 data can also be integrated into existing third-party performance monitoring and visualization tools.

ANALYTICS

The O8 platform generates millions of data points including traffic patterns and activity within a space; mobile activity; and commerce transactions. Making sense of these data points and combining them with other outside data streams—such as calendar events and weather information—turns data into valuable business insights leading to the identification of opportunities to increase bottom-line results. O8 data can also be integrated into existing third-party analytics platforms.

IOT DEVICE MANAGEMENT

There are thousands of devices ranging from beacons and sensors to connected portal displays able to support the enhanced Guest experiences enabled by the Orchestr8 platform. Many of O8's strongest use cases take advantage of multiple groups of these IoT devices working in concert within the ecosystem. The Administration Domain provides a secure and logical way to manage and monitor these devices.
The Services Domain is the heart of O8 and where most of the system capability resides. It is comprised of three layers which build to a pinnacle of exposed APIs allowing internal and external systems to integrate and communicate with each other. This approach enables the development of custom rich ecosystems specific to each installation’s requirements and aspirations.

THE INFRASTRUCTURE LAYER

Orchestr8 is supported by multiple systems that get to know each other while also getting to know the Guest. This Infrastructure Layer includes the relational, NoSQL, geospatial, and map oriented databases one might expect. The O8 infrastructure also includes our networking and communication components and our IoT subsystem. The IoT component manages input from a variety of sensors as well as providing administrative functions for provisioning, monitoring, updating, and servicing this hardware.

Less obviously, however, the Infrastructure Layer also includes all of the legacy systems from which we draw information. This can involve client databases, CRM data, purchase histories, and any other source of information that provides context and narrative to guide the Guest’s experience.
THE CONTEXT LAYER

This layer is what distinguishes the Orchestr8 platform from general purpose software. The Context Layer gathers and manages information relevant to a Guest’s interaction with an O8 connected ecosystem.

While contextual information can be applied to various different entities, it is most often relevant to people and places. Context for a Guest, for example, might include information such as:

- Personal details: name, age, gender, address, etc.
- Journey details: What is the Guest trying to do and when did it/will it happen?
  - Time period for the journey
  - Sequential elements of the journey, for example, required steps
  - Scheduled activities, if applicable
- Sentiment
  - Customer Experience Index (CX Index)
  - Notable events
- History
  - Previous purchases
  - Previous journeys
  - Detailed customer activity events
  - Customer feedback
- Genome
  - Segment membership list
  - Metadata tags
  - Customer-specific models

Beyond the personal details, the Guest Context informs Orchestr8 services of what the Guest is trying to do and how he/she is feeling. At a hotel, for example, a Guest may be at the start of a 3-day stay (the “journey”) and may be feeling “bad” (negative CX Index value) because his/her luggage was lost by the airline. This context would help the hotel staff both to craft their interactions with this Guest and to offer relevant services (e.g., replacement toiletries).
A Guest’s history and genome are data that are generally used for analytic and predictive purposes. This information helps to provide answers to questions such as “what activity might appeal to a Guest?” or “What restaurant should we suggest?” or “Is this Guest likely to buy this sweater?” Orchestr8 contains components to facilitate gathering this information from legacy systems and from machine learning sources.

For a place, context might consist of information similar to:

- Venue details: name, description, floor plan, etc.
- Journey details
  - Time period for the journey
  - Scheduled activities
- Venue history
  - Previous journeys
  - Detailed customer activity events (taking place in venue)
- Venue genome
  - Segment membership list
  - Metadata tags
  - Venue-specific models

The specific details of the Context Layer depend greatly on the legacy systems that are available for integration with Orchestr8 and the degree to which our clients have or want to invest in analytics and machine learning. It is difficult to provide off-the-shelf solutions to these problems as they vary greatly by industry and even within industries.
THE SERVICES LAYER

At the pinnacle of the Services Domain is the Services Layer that exposes the combined functionality of the base Infrastructure and Context Layers to consumers of those capabilities.

The primary value of the Orchestr8 platform is to provide an ecosystem where wildly disparate capabilities are brought together to be accessed in a uniform and standardized manner by any number of possible contributors and consumers. The Services Layer is what enables these subsystems to interoperate harmoniously.

The Services Layer can expose functionality that is exclusively internal to O8 such as Level 11’s proprietary Lok8™ Location Services module. Alternatively, services can be integration middleware that unlocks access and interoperability to external third-party systems.

The Services Layer permits three axes of communication:

- Internal communication between module components
- Service-to-service communication between modules
- External communication with consumers of Orchestr8 functionality through exposed APIs

Internal communication permits functional components of a single module to communicate, while service-to-service communication between modules bridges inter-module connectivity in a standardized manner. The ability Orchestr8 has to expose APIs to external systems unlocks powerful opportunities for advanced integration scenarios that can serve as important differentiators from competitors.
CORE SERVICES

While Orchestr8 contains a number of foundational services, two are worthy of a more detailed description: Identity and Location Services.

IDENTITY SERVICE

The Identity Service pulls together a number of related aspects into a holistic view of a Guest in an Orchestr8 environment. By integrating these dimensions to a Guest and their relationships to others and the environment, O8 gives powerful insight into their experiences and what they need, want, like, and don’t like.

AUTHENTICATION AND AUTHORIZATION

Authentication and Authorization (A&A) are two related but very different concepts. Guests authenticate into O8 by providing credentials proving they are allowed access to an account in the system. Once authenticated, each Guest will be authorized to access resources determined by their privilege levels in the system. Orchestr8 is designed to work with multiple different A&A technologies, allowing deployment of the right-sized approach for a client’s needs.

GROUPS AND FOLIOS

No person is an island, and deep understanding of interpersonal associations is key to gaining the full picture of an individual’s experiences. At the lowest level, Orchestr8 uses location data and machine learning techniques to identify patterns of activity that identify relationships between individuals. At the highest level, O8 provides advanced social tools for explicitly creating group relationships and interactions with others.

One powerful capability utilizing O8 groups is folios. Folios permit groups to share both means of payment (such as credit cards on file) as well as entitlements (such as tickets and coupons). Folios most commonly are the means by which families can enjoy simple and easy payment management for goods and services for children when they are not with their parents.

THE GUEST GENOME

The Guest genome was discussed previously in the Context Layer section above. It is in the Context Layer that the raw data is collected and analyzed and where new insights and understanding emerges. However, it is the Services Layer that enables raw data to flow in and insights to flow out to internal and external consumers.

LOCATION SERVICES

Location services is at the core of Level 11’s domain expertise and intellectual property and is a central offering of the Orchestr8 platform. Starting with our work at Disney on the Magic Band program and the XConnect platform, Level 11 has demonstrated world-class leadership in creating connected, location-aware ecosystems.

Location services is a broad and diverse topic that encompasses an extensive range of manifestations. This section will explore the important options available when investigating the right location services capabilities for your situation.
LOCATION SERVICE BENEFITS FOR INSTALLATION OPERATORS

GUEST SATISFACTION
By investing in location services, a space is immediately brought into alignment with twenty-first-century expectations by Guests. Almost every Guest will have experience navigating to a location using these technologies. To have them left at the door feels disjointed and out of touch with the experiences just outside.

Properties that do supply location services will, therefore, be perceived as aware and concerned about providing the experiences that the Guest has grown accustomed to in the modern digital world.

ANALYTICS
Knowing about an individual allows for customized experiences and improvements. However, big data location services provides a path to knowing how to make many people happier all at once.

Analysis of traffic patterns and activity in the space can result in valuable insights. These insights can enable positive and dramatic changes for all Guests and identification of opportunities to increase bottom-line results as well.

A key example of leveraging location data has been to matrix it against other data streams such as weather and calendar related events—namely school closings, holidays & sporting events. Predictive analytics and machine learning applied to these aggregated data patterns have subsequently identified opportunities that otherwise would remain invisible. These opportunities have been harnessed to trigger incentives to Guests and decisions to participate in day-of ad campaigns that drive significant uplift to unplanned attendance.

These opportunities are a multiplier on the return on investment for location services and should be explored early to understand what feature sets could unlock new avenues to growth and Guest satisfaction.

LOCATION SERVICE BENEFITS FOR GUESTS
Although there are many potential entanglements of location services with other kinds of features, for Guests there are two primary use cases of location services—search and wayfinding. Together these combine to be a powerful unified tool enabling Guests to maximize their experience of a physical space by quickly finding the objects and services they want and discovering where they can be found.

SEARCH
Search is a tremendously broad concept and can be focused on many different attributes of a search term. In the context of location services, search is inherently oriented to finding where in the world something is. Guests explicitly engage with location services primarily to find Points of Interest (POIs) which come in many flavors—places, products, and people being primary among them.

Location-based search results are, of course, typically displayed and most commonly useful on a map, although a list of results by itself could be sufficient. Hybrid approaches integrating both (when there is sufficient display area) are even better.

Once a POI is found, the next mission is assisting the Guest with getting to them.
WAYFINDING
Features related to navigational information to POIs is known as wayfinding. Wayfinding comes in many flavors based on context and available technology. Below we explore a number of progressively more sophisticated wayfinding features and approaches.

STATIC MAPS
At its most trivial, wayfinding technology can simply be supplying the Guests with a map. A slight improvement on this rather minimalist approach is the addition of a “You Are Here” dot and label. Though in modern terms a map is not a strong offering, it has sufficed for most of human history and should be noted as the baseline for modern wayfinding to improve upon.

STATIC ROUTING
The next level of sophistication above Babylonian technology is the addition of a route annotation to the map showing the best path from one point to another.

This feature can be as general as allowing the Guest to choose two (or more) arbitrary points on a map and having the wayfinding system find an optimal path. More typically useful is a system that knows where the Guest currently is and provides a search mechanism to find a POI. The wayfinding system then creates a custom map with a static route from the current location to the selected POI and delivers this to the Guest.

This level of capability is an acceptable modern Wayfinding experience that, while requiring only a minimum of physical infrastructure to support, provides a very useful experience to the Guest.

BLUE DOT WAYFINDING
Blue Dot Wayfinding (named after the standard “You Are Here” blue dot on most GPS-enabled maps) brings the expected level of wayfinding capability to indoor navigation. In order for this capability to succeed, spaces must be instrumented with technology that can work with location services enabled maps to pinpoint interior locations.

There are a number of possible approaches, but the standard options for Blue Dot Wayfinding are Wi-Fi and Bluetooth. Each technology has its strengths and weaknesses. Wi-Fi is nearly ubiquitous but is also typically very inaccurate. While not as pervasive as Wi-Fi, Bluetooth is sufficiently common to have some utility in a similar way. However, Bluetooth Low Energy (BLE) (a specialized standard of Bluetooth) is especially well suited for Blue Dot Wayfinding with regards to power use, standardization, and accuracy. Orchestr8 employs BLE as the currently preferred technology for being able to most accurately pinpoint someone indoors with a mobile device and location services enabled map.

DYNAMIC BLUE DOT REROUTING
A feature often assumed by the Guest but not always implemented for custom indoor location services maps is dynamic rerouting if the Guest deviates from the previously plotted route. This feature is not a significant technical increase in complexity over “static” Blue Dot routing as it does not need any additional hardware or software capabilities in the system.

Including live update capabilities will create an additional tax on the network and server traffic which should be accounted for in overall service planning and deployment. However, this feature is a significant advancement in the experience over a static route, so not including this capability should be considered very closely as its absence will most likely result in Guest dissatisfaction with the feature.
DIRECTIONS

A final feature that falls under the wayfinding umbrella is the humble direction such as "turn left at Albuquerque". This can be considered a form of routing.

At its most mundane, directions can be a list of navigation steps to take. However, when augmented with other technology, they often can be the preferred means to do wayfinding. For instance, audio just-in-time directions can be a lifesaver when driving solo in a new or challenging environment.

Text can also help annotate a route with useful and possibly interesting information that is otherwise difficult to add to a map. One example is communicating the distance to the next directional change point. On a map, a long leg of a trip can easily extend off the map, giving no idea of distance or time until the next change in direction. In text, the simple addition of how long a leg of a journey is can greatly ease a driver’s mind, reassuring him that he can not worry about navigation for a significant time period and instead just relax for awhile.

With these perspectives on wayfinding in mind, let's now explore the environments that indoor location services operate in.

LOCATION-ENABLED ENVIRONMENTS

Not all location-aware environments are created equal. The specifics of the physical infrastructure greatly determine the range of features and services it is possible to deploy. Next, we will explore the three big types of location service deployments and discuss their strengths and limitations.
YOU ARE THE BEACON

In the You are the Beacon configuration, the environment senses a device that identifies a Guest and his or her movements. Typically this is an active device (which means it has a battery) such as a BLE beacon or RFID tag in the possession of the Guest. This device sends an encoded signal to sensors in the environment with a unique ID keyed to his or her account. These signals are then relayed from the sensors to the location services system for processing.

This scenario has the biggest opportunity for collecting high-fidelity information about Guests and their behavior as the data stream is continuous, powered and identifying. As such, the system can perform detailed analysis on a Guest’s movement in the space, associated travel parties, personal interactions, and overall interests. This information can then be stored in the Guest genome for use in future interactions.

The Guest genome unlocks the potential for giving the Guest a rich and highly tailored experience. This capability can result in increased customer satisfaction due to the ease of locating what the Guest knows they want as well as receiving recommendations about what the Guest wants but is unaware exists or is possible.

While this arrangement allows for the richest data and greatest control, it is by far the most uncommon. It involves the willingness of the Guest to manage a tracking device of some kind which is inherently problematic; Guests may be uninterested in keeping the item in their possession at all times or they may have concerns about privacy and being monitored. Therefore this is an unusual situation that is typically only implemented in very specialized circumstances where the Guest is highly motivated to participate.
YOU ARE THE SENSOR

In contrast to the You are the Beacon scenario, You are the Sensor is the complete technical inverse. In this situation, rather than the environment sensing a beacon in your possession, a mobile app on your phone or tablet senses beacons in the environment and forwards that information to location services.

Typically this scenario occurs when the Guest installs the app when she wants to find places or POIs in the space or other kinds of information. This scenario has three very important differences from the previous one:

• The Guest doesn't have to have possession of a beacon but does need to have a mobile device and install an application that talks to the location services
• Data collection happens only when the app is running
• The Guest can benefit from location services through search and wayfinding UI on the app

Additionally, the Guest can be either authenticated into the app or not. If logged in, the app can report who as well as where the Guest is and add data to the Guest genome.

With regards to the “logged out” user experience there are two possibilities:

• If the Guest has logged in at least once but is currently logged out, the app has a policy decision to make. It can either report data as the last Guest that was logged in or can report as an anonymous user. Most likely the Guest would expect their privacy to be respected and to not have their identity broadcast, but that is up to the app creator to implement.

• If the Guest has never logged in to an account, the app can not send the identity of the Guest but can actively send anonymous position data.

Motivating Guests to install the app is the biggest challenge for this scenario. Finding the right methods to 1) build awareness and 2) provide adoption incentives are keys to success.
DEVICE-AWARE ENVIRONMENTS

Unlike either of the previous scenarios, device-aware environments passively monitor for wireless devices of all kinds without the participation of the Guest. This technique is possible due to the information sent by devices searching for Wi-Fi access points or with Bluetooth radio enabled—even when not actually connected to any other system or device.

This scenario is almost always anonymous but is very useful for gaining insights into the aggregate behavior of people in the space nonetheless. These insights can be used to identify deficiencies in the ergonomics of the physical space to identify bottlenecks, estimate wait times or inspect standard patterns of use.

When overlaid with visualizations such as heatmaps or metadata such as product placements, weather conditions or special events, deep knowledge emerges that can serve as the basis for increasing Guest satisfaction and retention and for designing new and novel engagement models.
The Experience Domain is the outermost layer of Orchestr8, where the Guest interacts with the capabilities of the platform, while the platform interacts with the environment and therefore indirectly with the Guest.

The most immediate user experiences are through web, mobile, and kiosk touchpoints. In addition, the adoption of sensors and wearables into a digital ecosystem unlocks exciting opportunities to engage with Guests in powerful new ways using the Orchestr8 platform.

The O8 platform is designed to give the greatest possible configurability and control of the user experience by the installation operator. Three aspects of the platform enable this:

- Modules
- Feature Unlocking
- Advanced Content Editing Capabilities

**MODULES AND THE EXPERIENCE DOMAIN**

As previously discussed, administrators can decide which modules are most appropriate for their installations and add or remove them from their deployments at will. Modifying these settings will make these features appear and disappear from the web and mobile experiences.

On-the-fly installation reconfiguration is a powerful feature and the possibility for Guest surprise could be high. Due consideration must be made for confusion that dynamic changes to the experience could create for the Guests who were accustomed to the previous configuration.
FEATURE UNLOCKING

Feature unlocking is a powerful feature of O8 that gives administrators control of the visibility and use of a feature in a fine-grained way. This approach can be utilized to keep features dormant until they are formally released, legally allowed, or fully paid for. Depending on the implementation, any arbitrary group of users can be selected, including down to the granularity of an individual user. This capability can be used to select groups related by different attributes, such as country, payment status, or role in a system.

As previously discussed, deployment of a module adds new, related functionality to potentially all domains of an installation—Administration, Services, and Experience. If feature unlocking is a capability of the module, then the Administration Domain will contain the management tools to access the capabilities deployed in the Services Domain and expose them in the Experience Domain to end users.

ADVANCED CONTENT EDITING

The purpose of certain O8 modules is to deliver content to end users—both on the web and mobile as well as on-prem devices and screens. Utilizing the advanced customization capabilities of modern Content Management Systems (CMSs), Orchestr8 enables the publishing of sophisticated and dynamic content to any target platform.

With our appreciation for the power of novel CMS publishing capabilities, Level 11 brings a creative panache to the discussion of how to surprise and delight Guests with novel experiences created using these platforms.

ADDRESSABLE AUDIENCES

The Experience Domain addresses two primary audiences: Guests and Host Staff. Each will be granted a set of interfaces and possible interactions that have been designed for them which will allow them to accomplish specific tasks. In many cases, these two sets of interfaces share a great many features (directories, maps, inventor displays, bookings, etc.), but the specific roles and responsibilities of each audience will dictate the final delineations.

Providing thoughtfully designed user experiences to both Guests and their Host counterparts allows for dramatic reductions of friction in Guest-Host physical interactions at all phases of the Guest journey. The ecosystem’s technology enables human service interactions which are the base product of the hospitality industry. The O8 platform is purpose-built to take a Host-first approach to maximizing a Guest’s experience. This is accomplished by removing friction in Point of Sale (POS) systems and making the capabilities of the Host property seamlessly accessible from within Staff apps.
A SAMPLE EXPERIENCE

In addition to Orchestr8, Level 11 also offers Experience Design Engineering (XDE™). XDE is our methodology for examining clients’ existing business processes and their customers’ journeys. XDE allows us to understand points of friction and to identify opportunities for using technology to eliminate these and replace them with positive interactions. Many clients face challenges that can further be understood by placing them within our PAER (Plan, Arrive, Experience, Remember) framework. This framework, while developed for the hospitality industry, works remarkably well in a variety of applications.

Our work on some of the world’s largest and most ambitious experience projects has given us deep insights into these problems. The following is an excerpt from current work in the retail hospitality industry that leverages our PAER Guest journey framework.

Mr. Kim is planning a trip to the NYC area with his family and discovers a shopping and entertainment complex. As he explores the venue’s website he discovers transportation options from their hotel as well as the mall’s amenities and events. The breadth of activities available at the mall convinces him to visit the mall for the day. Mr. Kim plans to purchase numerous items to take back home given the favorable exchange rates and tax-free location of the mall. He notes that there are plenty of options for the kids to explore while the parents are shopping. He decides to purchase advance tickets to Cirque du Soleil for an evening show.

In the checkout process for the tickets, Mr. Kim is offered the option to become a VIP member by creating an account. The tickets are sent to his iCalendar and Apple wallet for ease of use. After creating his account, he is rewarded with VIP entry into the show enabling him to skip the main line at the box office. The account confirmation reminds him to download the app to take advantage of his VIP entry to the show.

Mr. Kim downloads the app. When he signs into the app, he is reminded of the event he has purchased tickets for. The app also suggests inviting family members to enable easier coordination during their visit. He sends invitations to Mrs. Kim and the two kids. The family members are automatically linked to his account and folio during the invitation process.
ARRIVE
On the day of his planned visit, Mr. Kim receives a push notification of the upcoming event along with estimated travel time from his hotel. He is offered a customized deal for discounted valet parking. Mr. Kim accepts the deal, and it is added to his folio.

As Mr. Kim approaches the parking lot, he is guided to the valet parking area by the digital signage. After he hands the keys to the attendant, Mr. Kim receives a notification on his phone that the valet claim check is in his folio.

EXPERIENCE
The Kim family enters the mall, receiving a first-time welcome message on their phones. The kids are interested in exploring the theme park which was highlighted in the digital signage at the valet stand. Mr. Kim opens the app for navigation to the nearest kiosk, where they purchase the wearable wristbands to the theme park. Mr. Kim opens the folio section of his app and enables the wearables for snack purchases while the kids are at the theme park.

Mr. and Mrs. Kim spend the day shopping and accumulate a large number of shopping bags. Near the end of the day, Mr. Kim receives a notification that there is a concierge service that can hold their purchases during the show. Mr. Kim elects to take advantage of the concierge service and the Kims are directed to the concierge station.

Mr. Kim receives a reminder that the show is scheduled that evening. He sets a pin on the map at a nearby restaurant for the kids to meet prior to the show. After a quick dinner, the family walks to the Cirque du Soleil venue using navigation provided via the app. Mr. Kim’s VIP status enables them to skip the line, and they walk directly to their seats.

After the show, the Kims begin to walk back to the valet parking station. Mr. Kim receives a pop-up notification that he can request the shopping purchases to be delivered directly to the car. Mr. Kim says yes. When the family arrives at the valet, their car is waiting for them with their purchases in the trunk.

REMEMBER
Mr. Kim reviews the transaction history in the folio, which includes purchases from the kids while they were at the theme park and the photos that were taken on their rides. Mr. Kim is glad to see that his kids had so much fun while he was enjoying himself as well. He purchases 3 of the photos to be drop-shipped to his home where he will have them as keepsakes of their international adventure.
As systems continue to grow in scale, respond elastically to variations in load, and change architecturally (i.e., from monoliths to networks of micro-services), it has been necessary to evolve both the organizational models and the technologies that are used to manage systems.

DEVOPS ORGANIZATIONAL MODEL

Level 11’s Production Engineering team is responsible for establishing and maintaining the reference patterns for how Level 11 builds, tests, deploys, configures, monitors, and operates software. This is not a vaguely defined “DevOps” team, nor is it an Operations team. It is a team of software developers who focus on technologies and practices to support smooth end-to-end lifecycle of large-scale software development projects. Members of the Production Engineering team works side-by-side with developers of O8 platform “experience” features, sharing responsibility for systems operations.

THE FOLLOWING SIX PRINCIPLES OF SYSTEMS DESIGN ARE THE CORNERSTONE OF LEVEL 11 PRODUCTION ENGINEERING:

SELF-SERVICE
Teams must be able to manage application configuration and deployment without dedicated operational personnel.

AUTOMATED
Systems must be deployed and maintained through automation or automatic behaviors.

TRANSPARENT
The configuration and operating characteristics of a system must be made visible and comprehensible to team members.

RELIABLE
The most feature-rich system is enormously devalued if it is not reliable. Reliability cannot be attained by any one discipline, but is the net effect of multiple disciplines operating in harmony.

INCREMENTAL
Delivering many small changes is preferable to delivering fewer large-scale changes.

CONTINUOUS
Fast-feedback loops from CI/CD result in higher quality code because issues are identified sooner (and are confined to smaller units of change); they also provide constant proof of the safety of the deployment process.
TECHNICAL CONCERNS
OF A MODERN PLATFORM

MICROSERVICES ARCHITECTURE
Level 11’s Orchestr8 Platform is developed and delivered as a network of microservices. Each microservice is a small, independently-deployable process (a “service”) that handles a set of related functions. In most cases, microservices communicate with other services using web protocols (i.e., HTTPS / Websockets).

Microservice architecture is rapidly replacing traditional monolithic architectures due to the strong benefits of loose coupling in a well-designed network of services. In contrast to monolithic architectures, the microservice architecture of O8P ensures that Level 11’s development teams have peak agility in delivering new capabilities. Because each microservice has an independent lifecycle -- only minimally entangled with the lifecycles of other Platform components -- our development teams are free to develop, test, deploy, and scale subsets of the Platform with autonomy and confidence.

SERVICE DISCOVERY
Service Registration and Discovery are critical elements of elastic, microservice architectures. In the Level 11 Orchestr8 platform, there are dozens on microservices, almost all of which can be scaled independently and automatically. It is entirely impractical to manage the inventory of all services through manual actions (for example, to re-configure load balancing or health checking when services start or stop).

Service Registration and Discovery is the foundation on which the O8 platform ensures automatic reconfiguration of the system, in response to normal elastic operation. At startup, all microservice instances in O8P are automatically registered in our distributed Service Registry. Service instances are deregistered when they are stopped or die. Other services listen for changes to relevant services and reconfigure themselves accordingly.

CONTAINERS AND CONTAINER ORCHESTRATION
In the past years, application containerization (esp., Docker) has reshaped the way that teams package, deploy, and manage software. Containerization has enormously increased the portability of applications by ensuring that applications are packaged -- in containers -- with all of their runtime dependencies. This model substantially decreases the operational burden of deploying software because Operations teams spend enormously less time installing and maintaining dependencies on the underlying host. And while containers don’t permit a limitless “write once, run anywhere” model, the reality is very close to this ideal (i.e., write once and run almost anywhere).

An enormous ecosystem of supporting tools and systems has grown up around the foundational containerization technology. Most of the tools in this ecosystem are focused on helping users leverage containers in large-scale development and runtime environments. After all, teams don’t want containerization as such; they want low-friction methods to deliver software with high-reliability, horizontal scaling, and transparent operation. Container Orchestration Engines, such as Kubernetes and Docker Swarm, provide critical capabilities in these areas, including abstractions that allow developers to deploy containerized applications in highly available clusters, with automatic load balancing, simple scaling semantics, log centralization, health management, and other features that are essential for production-grade operations.

Level 11’s preferred container/runtime technology is Pivotal’s Cloud Foundry, which is a Platform-as-a-Service (PaaS) that blends powerful Container Orchestration capabilities with a suite of higher-order features to support low-friction, self-service software deployment and management.
The Cloud Foundry platform can be deployed on any of the major cloud IaaS providers, including AWS, Azure, GCS, plus VMWare vSphere for on-premise infrastructures. For components that don’t fit the Cloud Foundry deployment model (e.g., databases, high-volume messaging technologies, object storage, etc.), these are deployed adjacent to the Cloud Foundry infrastructure via automated tools (i.e., Pivotal BOSH and Chef).

**HIGH-AVAILABILITY**

For Production operations, all components of the Orchestr8 Platform are deployed redundantly across availability zones. This is true at the level of the infrastructure foundation of the Platform (e.g., Cloud Foundry components, Kafka brokers, Zookeeper nodes, and other big-data technologies) and also at the level of O8P microservices that are running on Cloud Foundry. Cloud Foundry is aware of availability zones and its application scheduler ensure that microservice instances are distributed across more than one zone.

**CONTINUOUS INTEGRATION**

Continuous Integration (CI) refers to the process by which each software check-in is built and unit-tested automatically. CI creates a fast feedback loop with the development team, allowing developers to understand whether small units of change are fundamentally breaking the software. Continuous Delivery (CD) is the natural extension of CI, wherein software artifacts that are produced by a CI pipeline are automatically deployed, tested as part of the larger ecosystem of related services, and, optionally, promoted to higher-level environments.

All custom components of the Level 11 Orchestr8 platform are built, tested, and deployed through automated pipelines. In this environment, our development teams have exceptional situational awareness about the basic stability the codebase (through the CI mechanisms), plus deeper insights into the effect of any software change (via automatic integration testing controlled by the CD pipeline). An additional benefit of a mature CD process is that changes are pushed into test environments in small increments, so that defects can be tracked to a relatively small changeset and resolved; this is in stark contrast to traditional development models, in which teams accumulate thousands upon thousands of lines of code before deployment, which makes it very difficult to isolate and resolve defects.

**MONITORING STATUS AND PERFORMANCE**

All Orchestr8 Platform components expose health and performance information, for consumption by external systems. Cloud Foundry continually polls the health-check endpoint of each service instance to assess the basic health of the instance. If a previously healthy instance becomes unhealthy, Cloud Foundry will restart/reschedule the failing instance. All changes in health information are logged by the platform and displayed on Level 11’s operational dashboards.

Each O8P microservice exposes its performance information via a Prometheus metrics endpoint. Prometheus is a metrics collection and alerting system that was designed for distributed, elastic systems. Centralized Prometheus collectors, called “scrapers,” are integrated with the Level 11 Service Registry to discover Prometheus metrics endpoints across the entire network of O8P microservices -- and automatically configure metrics scrapers as microservice instances are scaled up and down. All performance metrics that are collected are persisted in a Prometheus time-series database, which allows Level 11 developers to query and visualize performance data of their services.

Additionally, Level 11 development teams build packages of integration tests that are safe to
run in Production systems. These integration tests run continually and are integrated with the Prometheus metrics collection system for centralized reporting and alarming (more on this below).

LOGGING AND ALERTS

In the Orchestr8 Platform, all infrastructure components and microservice logs are aggregated in Elasticsearch. Microservice logs are all structured as JSON messages, which allows us to leverage the powerful indexing and query functions in Elasticsearch (versus relying on naive full-text searching).

Level 11’s development teams build and maintain operational dashboards for all of their components in Grafana, a powerful visualization tool. Grafana is Level 11’s preferred dashboard/visualization tool because it allows our teams to visualize data from many different source systems, including Elasticsearch, Prometheus, Postgres, Amazon CloudWatch, and others.

Level 11 uses two primary tools to generate alerts on O8P health and performance: Prometheus and Elastalert. Teams define performance-related alerts via configuration of the Prometheus server. Alerts that are generated by Prometheus are routed through the separate Prometheus Alertmanager component, which handles deduplicating, grouping, and routing of alerts. The Alertmanager also handles silencing / suppression of alerts. Alerts are routed to Level 11’s incident management system, but may also be duplicated across multiple channels (e.g., email, Slack, or a customer’s incident management system).

For alerts against logged messages (i.e., messages stored in Elasticsearch), teams define alerts using Elastalert. Elastalert executes queries against Elasticsearch indices and evaluates the response according to one of Elastalert’s many rule-types. These alerts can be quite powerful and go well beyond the ability to alert on any single message matching a pattern (although that functionality is supported). For example, Elastalert can monitor quantity of messages matching specific patterns within a rolling time-window and generate an alert when the number of matching messages increases -- or decreases -- by a certain factor within a rolling window of time. Alerts generated by Elastalert are sent directly to Level 11’s incident management system.
While each component as installed today has been highly tailored to existing clients’ individual business-driven use cases, the myriad applications possible for each new installation remains both flexible and extensible. The value of Orchestr8’s powerful and dynamic platform is in having many of the hard problems already solved without sacrificing the ability for Level 11 to harmonize the specifics of new experience ecosystems to the brand values that a business desires to embody.
In addition to Orchestr8, Level 11 also offers Experience Design Engineering (XDE). XDE is our methodology for examining clients’ existing business processes and their customers’ journeys. XDE allows us to understand points of friction and to identify opportunities for using technology to eliminate these and replace them with positive interactions. Many clients face challenges that can further be understood by placing them within our PAER (Plan, Arrive, Experience, Remember) framework. This framework, while developed for the hospitality industry, works remarkably well in a variety of applications.

Our work on some of the world’s largest and most ambitious experience projects has given us deep insights into the problem.
Level 11 is a Seattle-based experience design and engineering firm that helps companies fundamentally transform how their customers engage, experience, interact with and remember their brand. Our team of designers and engineers follow a proprietary Experience Design Engineering (XDE™) approach, which addresses every touchpoint in a client’s connected ecosystem—physical and digital—across the lifecycle of users, Guests and customers.

Level 11 can be your technology, design and development partner across all stages of the product development and service processes.

For more information, visit www.level11.com.