ORACLE
JAVA EMBEDDED

Java and the Internet of Things: The Intelligent Platform for Healthcare

ORACLE WHITE PAPER | MARCH 2015

ORACLE

Java and the Internet of Things in Mobile Health

Expenses for healthcare in general continue to rise, especially for those with chronic disease. Healthcare providers are searching for ways to control costs while increasing the quality of patient care. Remote patient monitoring, part of mobile health (mHealth)—where chronically ill patients are cared for outside of a hospital or other facility—is one promising solution. Remote monitoring solutions also allow patients to be discharged from hospitals sooner, as their health can continue to be closely monitored from home.

At the end of 2013, there were three million patients using connected home medical monitoring devices worldwide. This figure comprises all patients that were remotely monitored by a professional caregiver ("mHealth and Home Monitoring," *M2M Research Series 2014*, Berg Insight). Additionally, it's expected that connected home medical monitoring devices will grow at a compound annual rate of 44 percent to reach over 19 million devices by 2018. Remote monitoring solutions' revenue reached $5.23 billion in 2013 and is expected to grow to $22.6 billion by 2018.

The promise of improved care with reduced costs is driving innovation in remote monitoring solutions. Together, healthcare providers and product vendors are working with Oracle Java Embedded to deliver solutions that include personalized medicine, improved healthcare management through advanced telemetry, and population health scenarios to better measure the effectiveness of care and treatment. Challenges include the need for connectivity, security, privacy, advanced analytics, data visualization techniques, and the remote management of monitoring devices. Oracle's Java and integrated Internet of Things technologies directly address these and other challenges, further improving patient care and reducing costs end-to-end.

The Value of the Internet of Things for Mobile Health

As more medical devices connect to the Internet, they become an important and growing segment of the Internet of Things (IoT). The term IoT is used to define a system in which the Internet is connected to the real world via ubiquitous sensors. The vision of IoT is to integrate diverse sets of data from physical sensors and the rest of IT to enable analytics that can anticipate events, issues, and other needs. As a result, the system as a whole can have a view of what's taking place at any location and any point in time. This vision leads to a world of connected systems that could greatly reduce waste, lower costs, and eliminate loss for just about any human-machine and machine-machine activity.

Given the daunting requirements of healthcare today (safety, regulations, security, privacy, and so on) and the rapidly emerging IoT technology wave, no other platform today is better positioned to enable an IoT strategy for healthcare than Java. With its ability to run on a wide range of devices from mobile and embedded systems with limited CPU and memory, to servers with immense power and capacity, Java powers a world of compute resources with ubiquitous connectivity.

Remote Patient Monitoring and mHealth

According to the World Health Organization, mobile health (mHealth) covers medical practice and healthcare supported by mobile devices, patient monitoring devices, and other wireless devices. It also includes applications for improved lifestyle and fitness that may connect to medical devices or sensors (fitness bracelets or watches, for example). mHealth can also include personal health guidance, health information and visualization, medication reminders, and telemedicine and remote patient monitoring via wireless communications.

Overall, there are five main segments of mHealth:

1. **Solutions for healthcare professionals:** including visualization and review of medical device data, patient history, population health statistics, care and treatment information, and other centralized healthcare support systems.
2. **In-patient monitoring:** the support and monitoring of patient healthcare within a hospital, acute or long-term care facility, or other care facility.
3. **Remote patient monitoring:** the support and monitoring of patient healthcare from home or other remote location other than a hospital or care facility.
4. **Assisted living and tracking:** for patients who live on their own but may require emergency assistance or tracking by family members.
5. **Personal wellness:** for otherwise healthy individuals who want to take an active approach to maintaining their health and wellbeing through diet, exercise, and other lifestyle methods.

Solutions across all of these mHealth segments create a connected care value chain involving four categories of service providers:

1. **Sensors and medical monitoring device vendors**. Device examples include medical monitoring devices such as blood pressure cuffs, blood glucose measuring devices, and integrated solutions such as smart hospital beds.
2. **mHealth connectivity solution providers**. These are cellular service carriers such as AT&T and Verizon in the US, and other manufacturers of specialized embedded communication devices. These devices may include Continua-compliant Bluetooth devices, devices that communicate via Wifi, and so on.
3. **mHealth care delivery platform providers**. Examples include data collection and aggregation platform vendors such as Oracle, Axeda, and other IoT technology vendors with a focus on healthcare.
4. **Monitoring service providers**. This includes specialized providers who offer integrated services to monitor patients and provide emergency response, if required, between device and communication vendors.

Across the mHealth segments and associated connected care value chain (Figure 1), patient mobility, device integration, and data analytics play a key role in adding value. The intersection of mHealth and IoT has uncovered a potentially huge area for cost savings within healthcare: monitoring and controlling the effects of non-infectious disease.

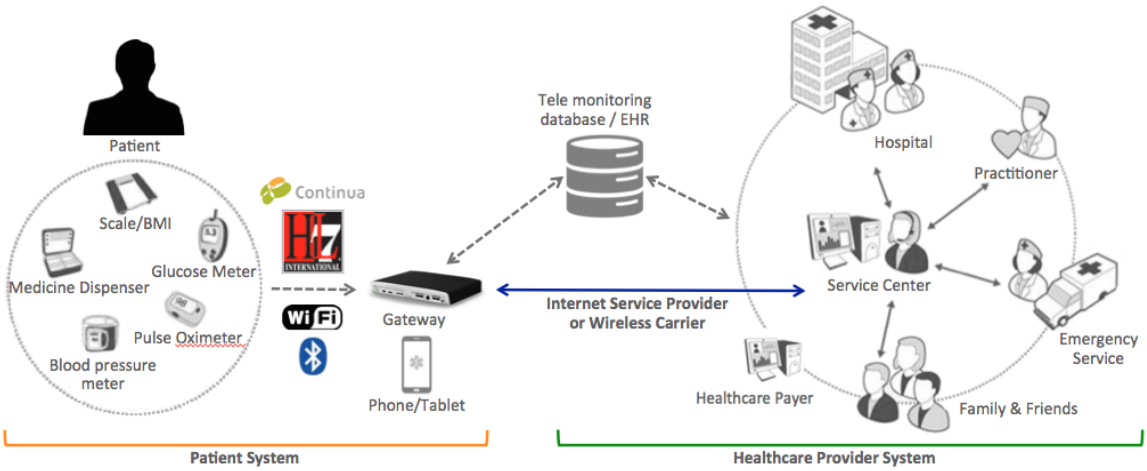

Figure 1. mHealth covers patient monitoring, mobile tools and communication, population health and analytics, and the payer.

The Rise of Non-Infectious Disease

The 21st century has seen an increasingly aging population, coupled with a rise in non-infectious disease-related health problems and deaths. Overall, non-infectious chronic diseases such as cardiac arrhythmia, hypertension, ischemic diseases, sleep apnea, diabetes mellitus, hyperlipidemia, asthma, and chronic obstructive pulmonary disease (COPD) comprise an increasingly large percentage of total healthcare costs.

For example, one third of all US adults are classified as overweight, and another third are considered obese. With over 400 million people globally classified as obese, weight-related health issues (for example, diabetes and hypertension) and related costs continue to rise. In the US alone, the estimated direct costs of hypertension are over $100 billion each year (Berg Insights). In addition, the prevalence of mild sleep apnea (often weight-related) is estimated to be around 13 percent of the population in the western world, whereas an additional seven percent are believed to have moderate to severe sleep apnea. Hyperlipidemia, a set of treatable conditions often resulting from a poor lifestyle, account for a global annual cost of between $300-400 billion.

Remote monitoring mHealth solutions are attracting the attention of healthcare providers globally as they have the potential to both improve the care and quality of life for patients with non-infectious chronic disease, while also reducing overall healthcare costs. These solutions involve patient self-measurement via remote medical devices, with reliable communication of the measurements and related data to a healthcare provider (such as a doctor, nurse, or technician) for remote diagnosis or analysis. As a result, patients maintain their independence and providers maintain personalized treatment while reducing costly office and hospital visits.

Revenue for remote monitoring solutions reached $5.23 billion in 2013 and is expected to grow to $22.6 billion by 2018 ("mHealth and Home Monitoring," M2M Research Series 2014, *Berg Insight).*

The Challenges of mHealth

Healthcare in general can be challenging and risky as people's health and lives are potentially at stake. Technology can help reduce risk and challenges in many cases (remote monitoring, automated alerts, and accurate medication dispensing, for example), but it can introduce new problems as well.

While gaining approval from regulatory bodies is usually the first step in bringing a new device or mHealth solution to the market, an additional challenge today is obtaining acceptance from reimbursement providers. To gain regulatory approval, a company needs to demonstrate functionality and safety. However, to gain acceptance from healthcare payers, a company needs to demonstrate the economic benefits of the new device or monitoring method. Although interrelated, we first need to understand these challenges individually:

- **Security, data safety and privacy:** Patient privacy and data safety are highly regulated and represent an area of liability to everyone involved. Therefore, both healthcare providers and solution providers need to find ways to ensure data security and privacy, including the security of remotely managed devices.
- **Implementation portability and cost:** Solution providers are challenged with the additional cost of connectivity. First, the cost of LTE cellular communications needs to be optimized. Second, many of the systems involved are embedded, traditionally requiring expensive specialized tools and implementation skills. As new medical devices are introduced, and advances are made in terms of remote patient care, everyone is looking for ways to cut costs through re-use and portability. This requires complex long-term planning for future mHealth device hardware upgrades to avoid re-creating systems due to evolving hardware. Standards and advanced development techniques are needed to offset added downstream costs and to remove portability concerns.
- **Enterprise integration**: The challenge is to find or build a platform that offers robust connectivity options, seamlessly connects to hospital or healthcare provider business systems, and remains cost effective over time.
- **Visualization:** There's a strong need to visualize data to convey information accurately and quickly, at a glance, allowing deeper inspection on demand as required. This requires integration with mobile devices, remote databases, remote analytics processing engines, and graphics systems with specialized algorithms.
- **Alarm fatigue:** Healthcare professionals are frequently bombarded with alarms and alerts, leading to a condition called "alarm fatigue" where alarms may not be noticed even when truly critical. The constant sound of alarms and noises from medical devices and patient monitors can cause staff to become desensitized to them. Alarm fatigue is present within hospitals, at-home care, nursing homes, and other medical facilities. The challenge is to establish alarm safety as a priority, help professionals identify the most important alarms, and create policies to manage alarms efficiently and properly. Further, increasing regulation in the area of alarm fatigue will require providers to add more complexity to their solutions.

The Advantages of Java in mHealth

Oracle Java and Java Embedded are key technologies for both the enterprise and embedded market, with billions of devices, gateways, desktops and servers dependent on Java to run their core functions. Java and Oracle's enterprise IT solutions work together to offer you a *single* end-to-end IoT platform for all of your mHealth applications, so you don't need to build custom infrastructure each time.

Oracle Java Embedded—optimized to run on resource-constrained embedded devices—packs enterprise server power into small controllers and devices. Whether it's used in a medical device, data aggregation and edge-analytics server, or other hospital or healthcare system, Java technology offers a number of key benefits (Figure 2) that make it the ideal platform for mHealth. For example, it enables headless, lights-out operation, a robust and secure application environment, remote software provisioning and management, and reliable end-to-end connectivity, all built on the industry standard Java language and virtual machine already used in IT. Overall, Java Embedded addresses all of the challenges listed and more.

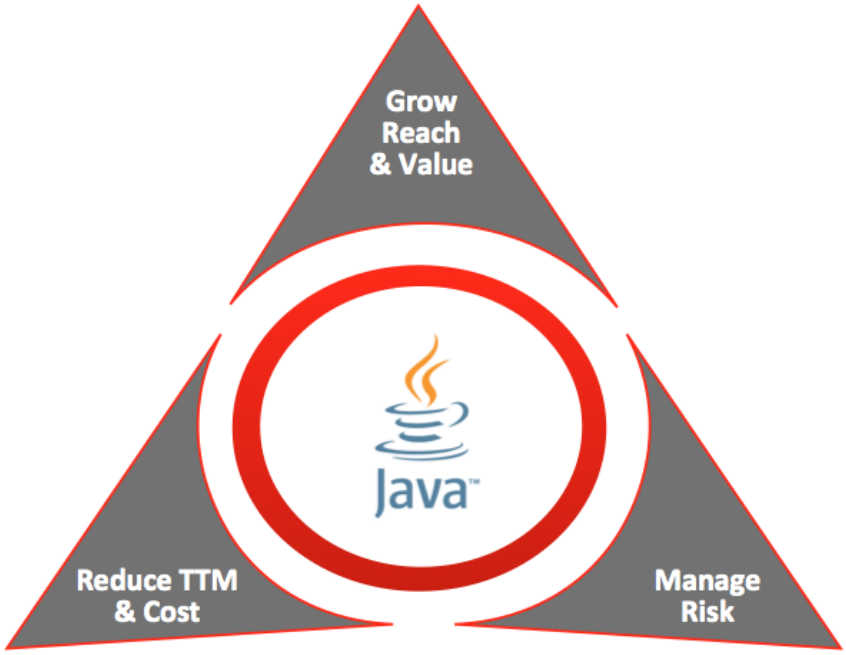

Figure 2. Oracle Java Embedded reduces costs and risks, and delivers value for end-to-end healthcare solutions.

Using Java, Oracle offers the ability to analyze events at the right location, in the cloud or on-site with embedded devices, depending on the relative value and time sensitivity of the data. The flexibility of Java Embedded is ideal for mHealth solutions, as it enables more processing and decision making to take place at the patient location. Localized analytics to process patient medical data at the point of capture leads to personalization, enhanced reporting, data enrichment, meaningful alerts, and remote patient communication and feedback.

Local, distributed analytics provide data aggregation and analysis, collection of patient feedback, and data safekeeping services for more effective healthcare management.

The flexibility of Java to run analytics end-to-end enables population health solutions, as patient data is combined with community health data, and visualized to offer insight and innovative solutions. Examples include advanced disease awareness, measuring the effectiveness of medical care and medication for disease treatment, as well as providing insight into factors that contribute to these diseases. Overall, by offering this flexibility, Java enables the true power in IoT: advanced analytics for data discovery, advanced remote treatments, population health and the prediction of patient illness.

Safety and Security

The Java Virtual Machine (JVM) is a proven secure technology, providing a secure environment to run multiple applications where each application is isolated from other applications, the operating system, and other software components (Figure 3). The JVM also abstracts the complexities of varying server and embedded hardware platforms, providing a consistent, reliable, and secure environment for your applications. Oracle focuses on security, with frequent patches and updates across varying server and embedded hardware platforms, so you can focus on your solution.

The security features of Java and its libraries include secure messaging, user authentication and authorization, device identity, data encryption, PKI, code signing, and more.

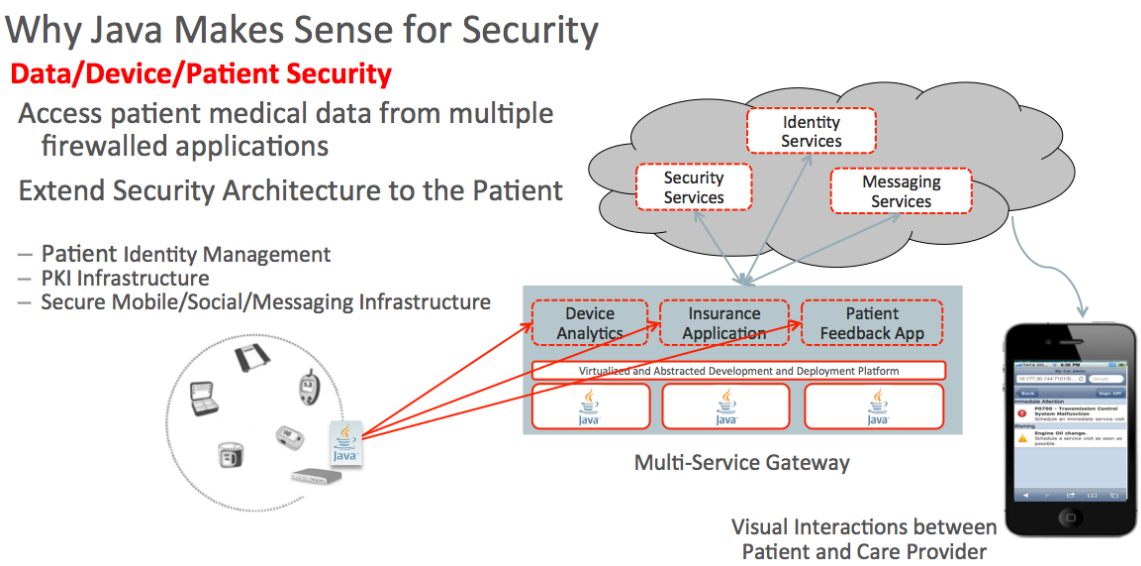

Figure 3. Oracle Java Embedded enables security and patient identity in mHealth solutions.

Java and other Oracle solutions provide the level of security needed to identify patients and their devices, and to secure their data to be sure it's not lost or tampered with in any situation. Additionally, Oracle provides the infrastructure needed to accurately and securely provision devices and communication gateways reliably to patients regardless of location.

Additionally, the "write once, run anywhere" model of Java development applies to security as well. Developing componentized Java applications according to supported security standards means you will be less likely to need new security measures as your hardware platform changes.

Total Connectivity Solution

There has been extremely rapid growth in remotely generated patient data, along with the need to send more data to the doctor and the patient. The importance of reliable connectivity will grow along with these data demands, and so will the risks if this data is lost. Java offers the robust communication services needed to reduce this risk.

Java's connectivity frameworks, along with the wireless communication and web service APIs, allow Java Embedded applications to work seamlessly, reliably, and securely with cloud-based enterprise services. This capability is crucial when deploying remote devices and infrastructure as part of an IoT value chain. Often, the connection type chosen is based on multiple factors, including transmission cost and device accessibility. Because Java supports a broad range of connectivity options, you can choose the type best for your solution.

The end result is that Java provides you with the processing and communication capabilities needed to integrate your mHealth solution, hospital records system, patient management system, and in-hospital solutions end-to-end. What's more, Java is engineered to work equally across your embedded devices, gateways, and IT servers, abstracting away the communication and hardware differences for ease of development.

Ultimate Flexibility

To maximize the benefits of IoT, solution providers require full flexibility in terms of where processing is performed. Java Embedded offers total flexibility regarding where analytics and processing take place, allowing you to change it later without redesign. This flexibility makes Java ideal for mHealth solutions as it enables more processing and decision-making to take place close to the patient, where medical data is captured. In addition, changes can be provisioned to upgrade functionality remotely or enhance inspection of a specific patient data point.

> *Using patients' own mobile devices as health hubs has thus far not been a viable option for remote patient monitoring ("mHealth and Home Monitoring,"* M2M Research Series 2014, *Berg Insight).*

Part of the Java innovation and value proposition for mHealth is its ability to quickly and securely provision new medical devices, service providers, and communication gateways in remote monitoring solutions. If your mHealth solution platform doesn't allow you to safely provision and personalize for patient-specific care, then you're putting your business and patients at risk if you cannot react quickly enough to changes in the industry, technology, and patient needs.

With Java, you can deploy new applications and update remote applications even after your solution is in use. This allows your globally deployed system to evolve to future needs over time via automated software updates. Further, Java Embedded supports remote device management, including the ability to update the JVM, libraries, and applications remotely, reliably, and securely.

Hardware Abstraction

The Oracle Java Embedded Virtual Machine offers an abstraction to underlying details and changes to embedded hardware, including medical gateways, remote medical devices, and integrated systems such as smart hospital beds. This allows developers to decouple your application from the many variations of hardware and operating systems in the embedded world. Existing Java applications can be deployed on new embedded systems with minimal or no changes. Developers just use Java and its single set of tools to focus on business value and time to market, regardless of the eventual hardware platform.

Optimized for Embedded

Oracle Java Embedded enables intelligent devices and distributed analytics to be developed rapidly and at lower cost, easily resolving many development and application issues. This includes easy integration with enterprise and internet-based environments. Oracle is focusing on helping the market reap the benefits of the transformation to smarter patient medical devices both in-hospital and in-home.

Maximized Innovation

According to software development rating and standards organizations such as TIOBE and the IEEE (Figure 4) Java is continuously ranked as the most widely used development language in the world. There are over nine million developers who use Java to develop a wide range of applications from enterprise to embedded. The ability to leverage Java's large pool of talent and tools better enables mHealth innovation and reduces costs via a standardized, end-to-end development environment.

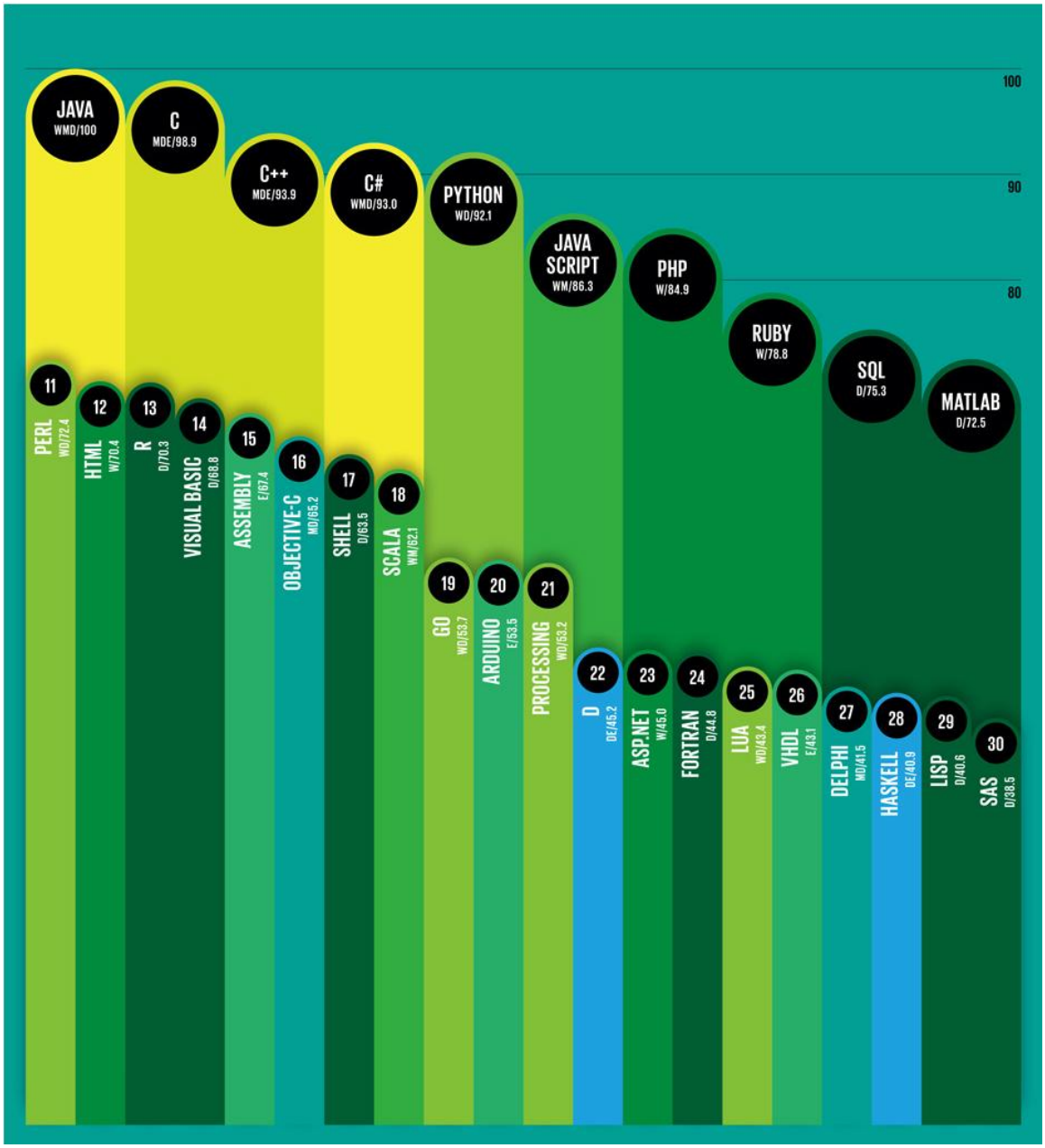

Figure 4. IEEE Spectrum's 2014 top-ranked programming languages.

Java provides the opportunity to have individual teams focused on medical devices and data acquisition, communications gateways, and back-end data collection and analysis using the same tools and environment. No other development platform can provide this unique set of values.

Java Reference Architecture for Remote Patient Monitoring

Oracle has defined a sample reference architecture to illustrate how Java Embedded can solve many needs in a remote patient monitoring solution. This reference architecture (Figure 5) serves as a guideline for the multiple use cases to which Java can be applied.

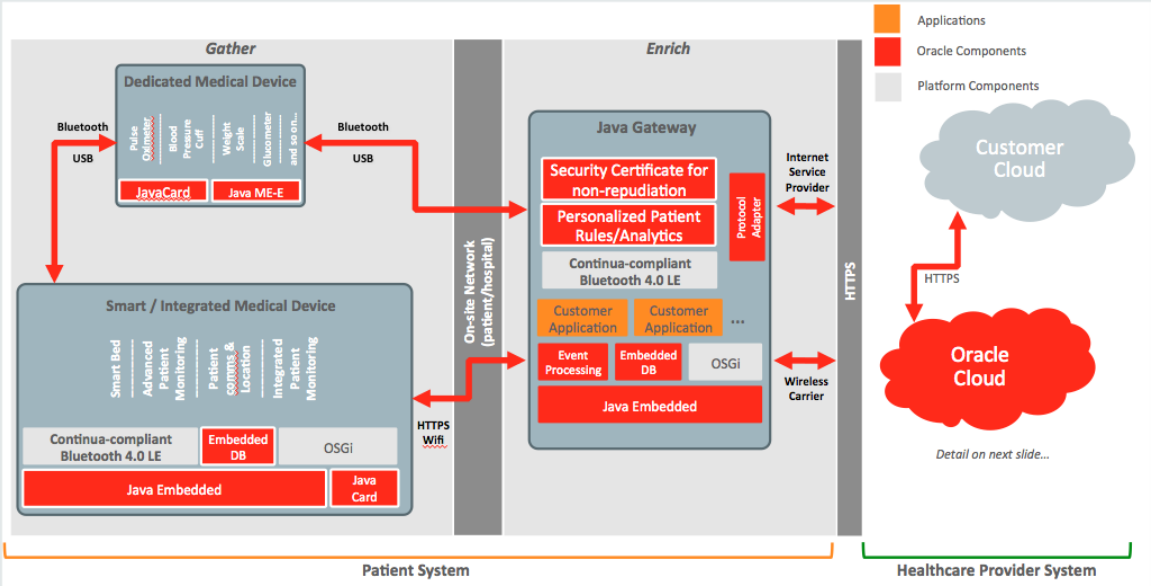

Figure 5. Oracle Java reference architecture for mHealth remote patient monitoring

The architecture covers and supports multiple applications within a single mHealth gateway, or distributed across multiple embedded computers and devices with high-speed reliable communications. Example use cases include:

- Remote monitoring of patient vitals
- Correlation and trend-based analytics to identify and signal potentially life-threatening scenarios for chronically ill patients
- Remote operation and administration of infusion pumps and other automated medicine dispensers
- Combining real-time patient data with historical data to gain insight into population and community health issues and resolutions

To ensure safety and security of the systems created and managed, Java Embedded can support healthcare standards set forth by regulatory bodies via the following comprehensive set of industry standard implementations:

- Certificate-based security, Java cryptography extension with crypto acceleration, and near field communication (NFC) support
- Java secure sockets extension (JSSE) for secure communication
- Java authentication and authorization (JAAS) for user, device, and data identity
- Public key cryptography standard (PKCS-11) for data encryption
- Security and trust services (SATSA) for additional encrypted security features and communication capability

Saving Lives with a Java-Powered mHealth Gateway

Incorporating an on-premise device gateway opens the door for local analytics that can be personalized for each patient and the monitoring devices they've been given. Specifically, Java and other embedded technology (such as a rules-based event processing engine) can monitor and correlate local device data in real-time to remove risks that exist in the current batch-based or manual data processing scenarios. Oracle Event Processing for Oracle Java Embedded is an example of a standards-based, Java-enabled event processing engine designed to work on constrained devices at the edge in IoT solutions.

For example, a slight elevation in a patient's blood pressure or weight, or a small decrease in activity individually may not raise any alarms. But viewed together, these readings may indicate the onset of acute heart failure or another critical situation specific to that patient's chronic health condition. The correlation-based local analytics can be personalized and augmented to prompt the user for other information (for example, the "how do you feel?" question) via a local tablet or other data-gathering device. As a result, the care provider is alerted to a critical health scenario with associated patient data in real-time, much sooner than before, to help predict and administer critical care before it's too late.

Additionally, the Oracle Java reference architecture supports integration with external systems through REST-ful services, embedded SOA-based Web services, and secure local storage of data. Through this architecture, Oracle and Java Embedded deliver the following mHealth business benefits:

- **Data management:** ability to filter, analyze, and correlate sensor data and take action on the large amount of data generated.
  - Real-time situational awareness, faster decisions, and immediate actions locally at the machine level and the enterprise back end
  - Agnostic of the event sources, destination or underlying communication layer
  - Tooling and event flow monitoring
- **Remote device management:** simplified and optimized network driven by remote device and application life cycle management.
  - Application distribution and lifecycle management
  - Real-time device monitoring
  - Service management and diagnostics
- **Embedded system diversity:** flexible architecture allowing for supporting applications running on a variety of devices from multiple vendors, minimizing application maintenance costs
  - Consistent, Java-based application development environment
  - Support for the range of devices from small embedded to enterprise-level servers
  - Application distribution and lifecycle management
- **Security:** critical data and applications secured on an open and integrated platform connecting devices to the enterprise network
  - Device enrollment and management
  - Access control, certification and compliance
  - Fine-grained entitlements and policy management

IoT Use Cases for Java in mHealth

Oracle is focusing on two main IoT use cases for Java in mHealth: acute care device management and population health with remote patient monitoring. Beyond these use cases and devices, end-to-end connectivity with Java and other Oracle solutions can lead to new, innovative, connected solutions where hospitals, doctors, nurses, family members, other caregivers, and insurance companies can more easily coordinate their efforts for patient care, with instant access to data regardless of location.

Acute Care Devices

Acute care is a branch of healthcare where a patient receives active but short-term treatment for an injury or illness due to an urgent medical condition, or as a result of surgery. Devices used in the administration of acute care include imaging devices, ventilators, smart hospital beds, infusion pumps, other automated medication dispensers or medicine administration.

Imaging Devices and Solutions

Imaging devices include magnetic resonance imaging (MRI), computed tomography (CT), positron emission tomography (PET), vascular imaging, x-ray, and ultrasound. Java Embedded can be used to monitor, control, and instrument imaging devices, such as MRI machines, not only to administer patient care and relay imaging data remotely, but also to gain critical insight into how the devices are being used and performing. Given the cost and complexity of these devices, the facilities that offer their use are limited, and those that do are often limited to only one of each device. Using Java and IoT to understand device usage and failure patterns can lead to predictive, scheduled, maintenance scenarios. In these cases, parts can be ordered, delivered and installed just prior to failure to prevent damage and to limit, or eliminate, downtime.

Invasive Patient Medical Devices

In-patient devices, such as ventilators and other invasive monitoring devices are used in more critical situations. Since reliable functioning and data capture are important, being able to actively measure many aspects of device performance while implanted can lead to more accurate patient monitoring and help avoid scenarios where faulty implanted devices need to be removed. Given that Java Embedded can run on small, constrained devices, it can be used to gather and transmit data on devices, or by executing on connected equipment external to the patient.

Non-Invasive Medical Devices

Java Embedded used with non-invasive monitoring devices, such as smart beds and smart medicine cabinets, enable data aggregation and correlation solutions. For example, a smart bed with an embedded gateway running Java can communicate with and gather data from other connected patient monitoring systems. All of these devices can be orchestrated and their data aggregated with a Java-enabled gateway, offering a single point of patient data gathering for the hospital, which greatly reduces complexity and related costs. Additionally, since the bed moves with the patient, a complete view of a patient's health can be delivered seamlessly, in real-time, even as the patient is moved around the care facility.

Another area Java Embedded and Oracle can enable innovation is in the area of infusion therapy (for example, insulin pumps). In this case, drugs are automatically dispensed and infused into the patient via a device, while the patient is outside of a care facility. Providing automatic, accurate readings (such as blood sugar) to the doctor helps to adjust the schedule of infusion, dosage levels, or both. The speed, reliability, and connectivity of Java can lead to finer-grained doctor feedback loops, and even closed feedback loops as devices self-adjust to real-time patient monitoring. More than any other development or runtime platform, Java has the built-in security and communication capabilities needed to bring innovative solutions such as this to market.

Population Health and Remote Patient Monitoring

Remote patient monitoring involves gathering patient medical data at the patient's location followed by remote analysis by a doctor, technician, or other care provider at a hospital or care facility. This solution is both cost effective and convenient as patients don't need to travel and care providers can monitor additional patients safely. Remote monitoring solutions enable patients—such as those with chronic disease—to be discharged from hospitals sooner, as their health status can be continuously monitored remotely.

Remote patient monitoring begins with the patient but involves end-to-end communication with backend support systems. These systems provide valuable visualization by combining real-time patient data with historical data, to allow the care provider to diagnose at-home patients and determine if they need additional instruction, changes in medication, or in-office or even in-hospital care. To facilitate these mHealth solutions, a Java-enabled gateway and other home-health devices offer tremendous value. Using Java, data can be collected, correlated, and analyzed locally and patient feedback collected if analytics indicate a potential acute health-related condition. Doctor or nurse feedback can be given to the patient remotely if device readings aren't being taken properly or according to a timely schedule.

Telecommunication providers may be called upon to provide reliable connectivity to remote patients, optionally store and forward data in case of device failures at either end, or provide secure communication channels and data delivery to ensure patient data non-repudiation. Java-powered embedded processors with built-in cellular capability allow mHealth applications to be built at lower cost than with any other developer platform. On the (IT) server side, live streaming data can be combined with historical data to provide views into potential community health issues.

Java-powered embedded processors with built-in cellular capability allow mHealth applications to be built at lower cost than with any other developer platform.

Population health studies intend to gain deeper insight into the causes of human illness, recovery, and survival rates. Many factors may contribute to illness, such as the economic status of the patient, geographic location, family and social status, as well as stress and other factors. The diverse methods of connectivity supported by Oracle Java Embedded allow for remote monitoring solutions deployed globally, which in turn contributes to large databases that can be analyzed as a whole.

Not only does population health and data analytics provide insight into the human factors behind illness, it can help measure the effectiveness of treatments, uncover economic barriers to recovery, and help discover more affordable and more effective treatments. Distributing the intelligence of device data gathering and the analytics of associated remote patient data using Oracle Java Embedded helps to distribute critical processing, lower associated costs, and maintain an accurate picture of the geographic influencers involved.

Getting Started with Oracle

The phrases “machine-to-machine” and “Internet of Things” are no longer just buzzwords; they’re proven key business and process differentiators. Oracle Java Embedded is the disruptive technology that gives providers the ability to innovate and meet the challenges modern healthcare brings, including security, reliability, modularity and updateability. Oracle Java Embedded is engineered and optimized to meet the unique challenges of designing intelligent devices and integrated solutions that unlock the business value of mHealth and remote patient monitoring.

An open, standards-based platform with an unequalled developer ecosystem, Oracle Java Embedded makes it faster and more affordable to get innovative, reliable, and secure mHealth solutions to market and provide the long-term support for success. Improved care with reduced cost is the heart of mHealth, and Oracle’s Java-powered fast data and edge analytics are becoming just as critical. Now is the time to utilize analytics to uncover value and treatment optimization in mHealth with Java.

To learn more, visit oracle.com/goto/javaembedded and oracle.com/iot or read the brochure, Oracle’s Internet of Things Platform: Solutions for a Connected World.

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200