Brandon Heck

brandon.heck@gmail.com
816.521.9194

Technologies & Tools

High Proficiency

- Java SE
- Java EE
- Web Services Development
  - Message-Driven Beans
  - REST (JAX-RS)
- JDBC, SQL
- Apache Maven
- JMX
- Jenkins

Working Knowledge

- Java Authentication and Authorization Service
- Java Message Service
- Java Native Interface (JNI)
- Websphere Application Server
- Apache Tomcat
- OAuth Protected Services
- XML
- HTML
- OSGi
- Linux Containers (Docker)

Basic Knowledge

- NoSQL
- PL/SQL
- JCache
- OAuth Providers
- JPA/Hibernate
- C++
- C#
- Groovy
- Eclipse RCP
- Ruby
- Clojure
- Node.js
- React.js
- Javascript ES6 & Node.js
- Shell Scripting
  - Bash
  - KSH
  - Batch

Responsibilities

- Gathering, creating, and assessing functional requirements
- Turning functional requirements into high-level architectural designs and low-level implementation designs
- Creating and assessing technical designs for use in high-throughput, highly concurrent systems
- Developing optimized, fault-tolerant software based on technical designs
- Ensuring that implementation follows the SOLID design principles
- Testing (Whitebox testing using Test Driven Development, Black Box Testing, Integration Testing, Performance Testing)
- Developing platform libraries to create a cohesive development, deployment, and software management ecosystem
- Adapting platform architecture to interoperate with new infrastructure
- Troubleshooting, debugging, and providing support to developers for general Java issues, issues involving open-source libraries, and issues related to libraries maintained by my team
- Mentoring new developers and developers new to the team
- Working cross-functionally with client support, infrastructure, and other development teams to diagnose production issues and identify fixes
- Work with infrastructure teams to prototype infrastructure tools and integrate tools with the service platform

Key Projects

- Enhance queue-based enterprise application service platform to run on SaaS infrastructure
    - Decouple runtime from legacy proprietary tooling
    - Create standard health check HTTP endpoints
    - Decouple service lifecycle from traditional domain lifecycle
    - Enhance platform for connections to multiple MQ queue managers
    - Implement MQ reconnection and service-side connection management
- Client, server, and middleware libraries which provide integration with existing tools and ecosystems
- Java EE Portlets running on IBM WebSphere Application Server (WAS)
- Reusable libraries to perform custom installation logic on WAS including data source setup, inspection and manipulation of environment entries, and setting up Websphere MQ resources
- Instrumentation framework (logging, counters, measurements, diagnostic context)
- Cache framework with extensible coherency model
- Object-Relational Model data persistence framework
- JDBC framework and utilities, including configuration, instrumentation, and failover
- Inversion-of-control framework based on the Service Provider specification
- Enhancing, maintaining, and writing reusable and stand-alone business logic

Brandon Heck

brandon.heck@gmail.com
816.521.9194

Activities

- **Kansas City Java User's Group — June 2017-Present**
    - Organizer and host
        - Organize locations, sponsors, and speakers
        - Promote meetups to drive membership and attendance
        - Manage web presence
        - Communicate to meetup members

Professional Experience

Cerner Corporation — February 2019-Present

- *Associate Lead Software Engineer/Software Architect*
    - Conceptualize projects necessary for maintaining technical currency, improving functionality, and delivering new functionality
    - Ensure technical feasibility of projects by researching, reviewing similar functionality in other applications, and creating proof-of-concept applications
    - Interact with vendor technical support resources to resolve issues, ensure technical correctness of usage, and add required functionality
    - Create and review high-level designs with multiple stakeholders and development teams
    - Use high-level designs to create and review low-level technical designs
    - Serve as technical lead for a team of 5-8 developers
    - Review low-level technical designs and code to ensure it meets standards for:
        - Technical correctness
        - Scalability
        - Supportability
        - Succinctness
        - Appropriateness of APIs
        - Extensibility
        - Functional completeness
        - Testability
    - Implement software changes

PayIt — November 2018-February 2019

- *Senior Software Engineer*
    - Create Java REST microservices
    - Use clojure to create a finite state automata for driving customer workflows
    - Enhance Node.js ES6 service to transform data for display
    - Integrate with client data systems
        - Explore existing client data model to find customer information for replacing legacy applications
    - Transform and store customer data models in MongoDB

Cerner Corporation — May 2008-November 2018

- *Associate Lead Software Engineer/Software Architect*
- **Cerner Java Developer's Meetup —** September 2014-November 2018
    - Organizer, Speaker
        - Coordinate topics and speakers
        - Emcee meetups
        - Speak on a wide range of Java-related topics tailored to how they are applicable to day-to-day development efforts at Cerner

- **Cerner Learning Facilitator —** March 2011-November 2018
    - *Millennium Architecture*
        - Cover the history, architectural overview, and concepts related to Cerner’s flagship product
        - Present purpose and usage information about operations tooling for developers
        - Lead laboratory exercises related to operations tooling and troubleshooting issues
    - *System Infrastructure and Services Development*
        - Facilitate material presentation and laboratory exercises for other developers on libraries used to solve common issues and standardize the operational model, including:
            - Standardized logging APIs
            - Caching framework
            - JDBC standardization utilities
            - Data marshalling libraries
            - Messaging middleware
            - Service specification APIs
            - Service runtime concepts and service containers
            - Best practices for services development on proprietary service framework
    - *Development Ecosystem*
        - Instruct developers on general development concepts and concerns, including:
            - Apache Maven
            - Software release best practices
            - Maintaining software assemblies
            - Strategies for API design
            - Automated testing best practices

Education

Master of Science
Applied Computer Science
Northwest Missouri State University
April 2008
GPA: 3.82

Bachelor of Science
Computer Science
Northwest Missouri State University
April 2006
GPA: 3.01

Continued Learning

Linux Containers and Docker - Pluralsight
Java 9 Modularity - Pluralsight