

**David S. Bakin**  
**Bakin's Bits, LLC**

[david@bakins-bits.com](mailto:david@bakins-bits.com)

<https://bakins-bits.dev/>

---

TECHNICAL STRENGTHS

Product engineering through all phases of product life cycle, focusing on *shipping product* with three primary characteristics: Correctness. Reliability. Performance.

Identifying and doing what needs to be done in the fast-paced limited-engineering-resources environment of the startup or product team.

Highly performant and highly reliable C++ (and Java) services/servers/libraries, via architecture, design, algorithms and data structures, attention to detail, and hard-won experience.

Design and development of concurrent (multithreaded) programs for multicore processors. Theory of concurrent systems. Debugging of concurrent systems. Low-level concurrency primitives and lock-free data structures.

Object-oriented design and development. Object-oriented architectures for applications and systems. Class and Object Framework architecture and development. Functional programming design and development. Service implementation via message queuing (e.g., RabbitMQ, Qpid).

Improving legacy software systems while maintaining production reliability and availability.

Expert in C++ (modern and 'classic'), Java, C#, C. Experience in x86 assembly. Compiler & development tool development.

---

WORK EXPERIENCE

**Hashgraph (originally: Swirls Labs)**

**2022—2025**

*Hedera Hashgraph*, Level 1 Blockchain

Responsible for EVM security, correctness, equivalence, and integrated features with the Hedera blockchain. Implemented Cancun upgrade and equivalence. Implemented system contracts and backend support for multiple Hedera features. Security investigations and resolution.

During major refactor (rewrite) of entire Hedera service (semantics of Hedera objects) by the entire Hashgraph service team: Suggested, prototyped, and drove the adoption of the successful testing strategy that ensured that the new implementation behaved exactly like the (then-)current production implementation. The new implementation was driven against the (persisted) events that the production mainnet had seen, and the record stream (block stream equivalent) was compared for all differences. Known (i.e., expected) differences were filtered out. All other differences were assigned for triage then fixing. Dozens of differences were found and fixed by the team, over several months, each one would have caused flaws in the Hedera semantics that would have been recorded forever, forever unchangeable, in the blockchain, had they been missed and gone into production.

## **Bakin's Bits**

**2020—2022**

### *BlockSifter*, Flexible and Production Quality Reimplementation of Bitcoin Core

Developing *BlockSifter* – a new open-source flexible but production-quality reimplementation of the Bitcoin Core node. Three goals: 1) Educational platform for learning about how a full Bitcoin node works (not just a blockchain verifier); 2) Showcase the engineering of a correct and reliable production-quality 24/7 service in modern C++ (C++17); 3) Provide an easily extensible Bitcoin full node that makes experimenting with new protocols and facilities easier than working in Bitcoin Core itself. Modular (plugin) architecture to make it easy to experiment with different implementations of each server-side "primitive". Functional programming techniques within C++ for data-transformation oriented processing with an asynchronous message-queuing backbone.

## **Xnor.ai**

**2019—2020**

### Machine Learning at the Edge – Platform and Systems Engineering

Implemented reliable and performant C++ gRPC-based services to provide an inferencing workflow in a critical highly-available production system in a datacenter environment, for a very large customer. Responsibilities included frequent delivery and working (remotely) with customer integration engineers to ensure our product worked in their demanding environment (and thus met contract requirements).

Added features to publicly distributed Ai2go machine learning/inferencing SDK (C, C++, Python). Added "introspection" to the inference engine (C++) so that running models could be queried for information related to configuration and runtime status. Exposed the configuration information through the SDK APIs. Building a "solution" capability for providing easy to use inferencing workflows involving multiple inferencing and image processing steps; a specific "solution" can be built – using the SDK APIs – and exposed to customers to address the use cases in various markets, simplifying the customer's evaluation and deployment of the product.

## **Bakin's Bits (independent consulting)**

**2020**

### **ECS Federal (follow-on to Xnor.ai project, after Xnor.ai acquired by Apple)**

#### **Object Tracking in WAMI (Wide-Area Motion Imagery)**

Specified functional requirements, designed and implemented a real-time object-tracking module as a drop-in replacement into a critical highly-available production system in a datacenter environment. Worked (remotely) with customer project managers to meet needs of their major customer. Delivered functional specification, design specification, high-performance (C++). Developed algorithms and methods tuned for the production data. Emphasis was on reliable operation and high quality tracks (as validated by customer).

Assisted customer software engineers and system engineers in troubleshooting, debugging, and suggesting fixes and improvements of existing production WAMI pipeline, as needed.

## **AIM Corporation (consulting)**

**2016—2018**

### **Meteorcomm**

**2018**

#### **ITCM – Real-time Highly Reliable Highly Available Highly Distributed Messaging**

Added features and solved defects for this real-time messaging system used in production by all United States Class I freight railroads, as well as most commuter and short-line railroads, in support of the Positive Train Control safety system. Linux-based C++ system running on ~50000 remotes (locomotives and wayside track equipment) and multiple back offices, over available transports of various kinds (220Mhz radio, cell, satellite, and direct wired connection).

Worked closely with SDETs to design/code/unit test for system testability. Provided training on unit testing techniques and coding for testability (especially for testing code with difficult external dependencies).

## **Expedia, Inc.**

**2016—2018**

### **Hotel Lodging Services – Search – Legacy System Support**

Improved reliability and diagnostic capabilities for a major legacy C++ system (20 years old), running on Windows (over 1500 boxes), responsible for over \$1B/yr in revenue. Reworked low-level thread pools (multiple!), web clients (multiple!), web API servers (multiple!) for reliability, correcting long-standing concurrency-related faults and reducing the service failure rate. Provided instrumentation that logged all unhandled C++ and Windows exceptions, and then fixed the uncovered thread deaths. Found and fixed issues with interactions with (multiple) external services. This work resulted in a reduction in service outage and in partial search result lossage.

Added a feature for filtering search results based on the customer's membership in hotel affinity programs, working with internal clients (brand front ends) to test and roll out the feature.

Supported live site problems and internal client tickets.

## **Bakin's Bits (independent consulting)**

**2013—2016**

### **iSpot.tv**

**2015—2016**

#### TV Capture/Commercial Detection/Transcode Service

Implemented a new message queue based service to orchestrate the capture of video and the analysis of it into fingerprinted commercials. A second generation of iSpot.tv's existing service, this service emphasized reliability and robustness with minimal operation/administration so that it could be replicated 200 times in 50 datacenters. Each service runs multiple sequential workflows, using a message queue (RabbitMQ) as a reliable, persistent, restartable state store as well as the inter-module communication fabric.

### **iStreamPlanet**

**2014—2015**

#### Aventus Live Video Transcoding

Implemented core product functionality related to real time video transcoding. Fully implemented SCTE-35 (broadcast industry ad signaling protocol) to handle national and local ads in video streams. Improved base (internal) framework functions in threading and event handling. Improved Aventus internal "plugin" architecture to enable better developer testing.

### **Mobisante, Inc.**

**2013—2014**

#### MobiUS Ultrasound Tablet

Implemented DICOM export and upload to PACS (image storage) capability in embedded Windows .NET application powering a dedicated ultrasound tablet. Provided performance and reliability improvements throughout the application. Refactored application for maintainability and to provide room for future enhancements. Finished an entire product cycle, including FDA process and documentation requirements.

## **Dynacron Group**

**2012—2013**

### Client Under NDA

Android (AOSP) system adaptation. Designed and implemented AOSP changes such that all of the Android system-provided UI (status bar, notification bar) – which is not provided, in Android, by a standard application – could be replaced by a single custom application in an embedded device, while still providing full Android functionality for standard applications. The replacement SystemUI would be in full control of the display, overlaid (with transparency) over standard Android applications.

### Client Under NDA

Android (AOSP) system adaptation. Designed and developed a system component that translated certain gestures into UI actions, so that all application UI components (Views, Fragments) could automatically behave according to the device's UX guidelines. Implemented AOSP changes to provide complete Activity/View lifecycle tracking to injected components without requiring changes to any application. Gesture tracking could be specified in XML, and styled.

## **Datacastle Corporation**

**2011—2012**

### **Endpoint Protection**

Implemented server (i.e. service) side and client functionality for the Endpoint Protection online (cloud) backup product. Scheduler service for reliably running and tracking repeating (maintenance and reporting) tasks on the service. New functionality for deleting and moving entities between companies and partners. Secure download verification of signed executable updates to the clients. Other product upkeep and maintenance tasks as required to meet customer requirements.

## **Amazon.com, LLC**

**2009—2011**

### **Fulfillment by Amazon**

Implemented the “Amazon Fulfilled Shipments Report”, for all FBA merchants, customized for all global regions. Implemented a data pipeline to move FBA ordering data from production databases through the data warehouse into reporting databases, combining and cleaning all shipment item data on the way. Optimized the queries for efficiency during the daily runs. Implemented the report using the FBA reporting framework. Validated all data pipelines, and report accuracy. (Thousands of these reports are pulled daily.) Subsequently implemented a separate data pipeline for China invoicing data. Implemented a new “freshness” mechanism (for all FBA reports) to show merchant, in the reporting UI, how recently the data was updated, reducing merchant calls for support. (SQL – Oracle dialect, Java, Perl)

Providing training, design reviews, and design support for the entire FBA team for data warehouse issues, including best practices for development production data warehouse applications to reduce operational overhead, and correct and efficient use of SQL to take advantage of (and not misuse) the special operational behaviors of the data warehouse.

## **Satori Software**

**2008—2009**

### **Mailroom Toolkit**

Analyzed US and Australian Post Office rule changes and implemented them in the address correction engines. (C++)

### **Mailroom Capture**

Implemented a keystroke capture facility so that user keystrokes could be displayed to user in a “symbolic” form, and then replayed into an arbitrary application at a later time. This involved capturing individual keystrokes and using a hidden window to discover how Windows would interpret key sequences including Alt-*nnn*, Alt-0*nnn*, and Alt+*nnn*. (C#, C++)

## **Microsoft**

**2006—2007**

### **Photosynth**

Wrapped low-level concurrency primitives. Provided an implementation of condition variables. Provided debugging features specific to the concurrency patterns used in the Seadragon Engine. Provided a protected work queue and task pool for the master/multiple-slave threading pattern. Diagnosed and fixed concurrency and asynchronous operation issues involving data races, deadlocks, and performance. (C++)

Implemented the “point cloud” 3D model view in Photosynth. Implemented an octree-based compression scheme for the point cloud raw data to speed point cloud loading from the internet. Implemented asynchronous loading and incremental display of the point cloud so that displaying the model’s point cloud did not interfere with the interactive performance of the image view. Optimized the point cloud display for efficient use of GPU memory (under DirectX 9) (as well as main memory). (C++)

Implemented frame-rate performance tracking, and frame-rate metering to achieve smooth interactive performance of the image view under varying load conditions. (C++)

## **Seadragon**

**2005—2006**

### Seadragon Phone Client

Ported Seadragon client code to Windows Mobile 2003 Pocket PC Second Edition Phone Edition. Replaced the application’s outer layer with Pocket PC specific code. Replaced the application’s low-level drawing methods with Pocket PC’s GAPI. Made a native Win32 implementation of the application’s abstract concurrency classes. (C++)

(Seadragon was acquired by Microsoft in February, 2006.)

## **Google Inc.**

**2003—2005**

### Google Pack

Designed and coded the mainline operations of the application, and was responsible for all security design and implementation of this security-critical application. Designed and implemented a module for querying a Windows system for the presence or absence of third-party applications – applications attributes were described in XML – used this to integrate multiple third-party applications into this application, tested it against 10 third-party applications. Designed and implemented a “hook” mechanism for tracking progress of (third-party) child processes, and monitoring changes to the registry and file system. Created the install, auto-update, and uninstall of the application. (C++)

### Google Desktop Search

Created from scratch the install, auto-update, and uninstall of the application, including OS-embedded components (BHO, deskbar, Layered Service Provider) – key attributes: reliability, small overall size, and speed. Implemented reliable multi-threaded components for asynchronous internet access to files and other services. Implemented many “foundation” classes providing access to various Windows technologies and services. Contributed to the performance analysis and optimization of on-disk data structures. (C++)

---

### PATENTS AWARDED

7,058,947	Resource Manager Architecture Utilizing a Policy Manager	June 6 2006
6,799,208	Resource Manager Architecture	Sept 28 2004
6,763,439	Disk Throttling and Priority Queuing for Streaming I/O	July 13 2004
6,748,443	Unenforced Allocation of Disk and CPU Bandwidth for Streaming I/O	June 8 2004

---

PATENTS FILED

US20200257960A1 Use Cases for Compressed AI Model

Feb 2 2019

---

EDUCATION

**Boston University**

Boston, MA

M.S., Computer Science

**Harvey Mudd College**

Claremont, CA

B.S., Mathematics