Located in Austin
Texas

Jeffrey Magder B. Sc. YC Alumni

University of Toronto

Cell Phone #: (650) 200-6009
Email: jeff.magder@gmail.com
Website: jmagder.github.io

Work Experience

Cofounder and Principal Engineer **Powder Finance (YC Backed)** **March 2023 – May 2024**

- Advanced and accelerated the data gathering process by creating Meetings AI, a GenAI bot that could connect to meetings across Zoom, Google Meet, Microsoft Teams, etc. and all other video platforms. On top of providing post-meeting summaries, you could talk with the bot by saying “Ok Powder”, and ask the AI questions about the in-progress meeting. The bot would respond using realistic sounding human speech based on the realtime transcript. (React, Python, Django, OpenAI LLM, Recall.ai, JavaScript, PostgreSQL)
- Created an auto question/answer system that listens in with Meetings AI, extracting important information and automatically populating answers to customer defined questions, if the information was mentioned in the video conversation, removing the need for manual entry for customer concerns and CRM purposes. (React, Python, Django, JavaScript, PostgreSQL)
- Auto-generated customer Mind Map visualizations through the auto question/answer system and Meetings AI, giving an efficient and detailed customer overview. (React, Python, Django, JavaScript, PostgreSQL)
- Implemented an AI powered system to parse, extract and ingest portfolio data from financial PDF statements, saving financial advisors dozens of hours manually entering data and allowing quicker turnaround for their customers. (React, Python, OpenAI LLM, JavaScript, PostgreSQL)
- Created systems for specifying and visualizing target allocations, and simulating the implementation details of buying/selling customer assets, measuring how it affects their risk profile and expected returns with respect to the target allocations. The system took into account the tax treatment of long term and short term gains as well as other underlying costs such as management fees. (React, Python, D3.js, JavaScript, PostgreSQL)
- Integrated with banks and investment firms using meld.io, for electronically pulling down customer assets and holdings, saving time and money for onboarding customers managed by financial advisors. (Python, React, JavaScript, PostgreSQL)
- Created a novel efficient frontier visualization, plotting portfolio holdings and their expected returns against their risk profile. This provided a quick way of showing if a given portfolio was taking on more risk without a commensurate increase in expected return. (React, D3.js, Python, JavaScript, PostgreSQL)
- Accepted into Y Combinator and ended the batch with $5mm raised.

Founding Engineer **Nonfungible Technologies - Arcade.xyz** **Apr 2021 – June 2022**

- As the first employee and founding engineer, I designed and created the Arcade.xyz application, originating its first loans worth $1.85MM (USD) 95 business days after hire. Continued loan origination and platform growth resulted in a successful Series A funding round of $15MM within 3 months. The funding round combined with the successful application work supported growing the company to 20 people over the next 6 months. (React, ethers.js, web3, GraphQL, JavaScript, MongoDB)
- Built the arcade.xyz parallel crawler that scanned all supported ERC20, ERC721, ERC1155, and CryptoPunk assets belonging to a given wallet address in under 2 seconds. (Node.js, ethers.js, web3, GraphQL, JavaScript, MongoDB)
- Built scaffolding for our web client, including cryptographically signed login. (React, MetaMask, JavaScript)
- Built the web3 application bridge, responsible for all on-chain smart contract interactions for all aspects of loan fulfillment including collateral deposits, withdraws, loan origination, payoff, default claims, etc. all wrapped in convenient React hooks. (React, ethers.js, MetaMask, web3, EIP-712, JavaScript, MongoDB)
- Built the Arcade.xyz server bridge, responsible for updating all off-chain databases in response to all smart contract events emitted throughout the loan life cycle. (Node.js, ethers.js, web3, GraphQL, JavaScript, MongoDB)
- Ran most releases, staged and migrated appropriately for zero downtime. (CircleCI, AWS, MongoDB)
- Reviewed virtually all PRs to our web client, bridge, and crawler repositories. (React, JavaScript, Solidity)
- Code reviewed the original asset wrapper, loan, promissory note, repayment controller and other smart contract work. (Solidity)
- Built a prototype ERC721 indexer using “The Graph”. It used a clever address based hashing scheme to parallelize indexing, working around inherent latency restrictions, bringing index time down to 7 days from 340 using a naive

implementation. Query times were brought down from 2 seconds to about 3ms. (Graph, GraphQL, AssemblyScript)

Senior Software Engineer **Barracuda FX** **Nov 2020 – Apr 2021**

- Maintained C# based bond and traditional financial instrument trading platform. (C#)

Senior Frontend Engineer **BitGo** **August 2017 – May 2020**

Multi-Currency Platform

- Added unified multi-asset support to BitGo's online multi-sig wallet interface, allowing for efficient overview, management, reporting, and auditing of multiple digital asset holdings. The brand new interface allowed BitGo to expand support to hundreds of coins and ERC20 tokens. (AngularJS, node.js, JavaScript, MongoDB)
- Simplified the BitGo API allowing for simultaneous queries across multiple coins, while still supporting legacy single coin API customers and providing a simple upgrade path. (Node.js, JavaScript, MongoDB)
- Implemented asynchronous web interface for Ethereum (and future ERC20 tokens) address generation, available gas, and associated fee address. (Node.js, JavaScript)
- Improved and developed BitGo's Royal Mint Gold (RMG) interface to production ready standards. (AngularJS, JavaScript, Node.js, MongoDB)

Onboarding, KYC and AML

- Created BitGo's first online KYC and AML verification process using Civic to support Royal Mint Gold's public launch. (AngularJS, MongoDB, JavaScript)
- Created BitGo's encrypted document upload service used for onboarding and as part of KYC and AML verifications. (AngularJS, MongoDB, JavaScript)
- Improved KYC/AML support with webcam and identifying documents upload, integrated with Jumio and Thomson Reuters for custody customers. (Angular, node.js, JavaScript, MongoDB)
- Helped develop back office software used for administering BitGo Custody Solutions. (Angular, node.js, JavaScript)

Billing

- Improved customer billing configuration processes and management tools, halving configuration effort, decreasing billing errors, and allowing for consistent on-time billing. (SQL, MongoDB, TypeScript)
- Implemented billing data analysis and auditing tools to investigate customer billing inquiries, and support data requests for the CFO. (SQL, Node.js, JavaScript)
- Designed a simpler more scalable data model to support efficient real time and retroactive analysis of customer transactional behavior, balance changes, and how they might be influenced by the greater market.

Maintenance and Craftsmanship

- Reduced build time using multi-stage docker builds, shortening the development cycle. (Docker)
- Increased productivity by modifying multi-stage builds to support live-reloading during development, while maintaining a consistent environment through production. (Docker, AngularJS)
- Improved product UX, fixed hundreds of bugs, and received a Craftsmanship award from CTO for the smooth staged rollout and polished look of BitGo's new multi currency interface. (AngularJS, JavaScript, Node.js, MongoDB)

Senior Web Engineer → Lead Engineer **Beckon** **June 2014 – July 2017**

- Designed and implemented **Beckon Dashboards**, Beckon's flagship drag and drop auto-sizing solution to marketing analysis. The project allowed differently sized-adjustable cards consisting of graphs, images, video and embeds to be laid out and maintained across resolutions while preserving aspect ratios. (AngularJS, CSS, JavaScript)
- Implemented multi-dimension drill down hierarchical dashboard analysis through the creation of, a lazy rendering multi-select widget capable of indexing, searching, and rendering itself in milliseconds, despite potentially millions

of entries. This replaced the original implementation which would take over a minute to render with large data sets. (AngularJS, JavaScript)

- Created **Beckon Scorecards**, a compact column based analysis tool designed to help spot trends and their changes across a plethora of metrics with just a glance. (AngularJS, JavaScript. PostgreSQL)
- Success from Dashboards and Scorecards lead to greatly increased product traction, resulting in large increases of data and data complexity. Implemented a customized differential compression scheme while redesigning the api to remove redundancy, lowering payload size up to 1500x from 90MB down to 60kB. (AngularJS, JavaScript)
- Exploited a new differential compression scheme to make multidimensional pruning searches cacheable, reducing response time on the most complex queries from 3 minutes to approximately 100ms or less. (AngularJS, JavaScript, Node.js, Redis)
- Upgraded dashboards to allow multiple simultaneous user edits using optimistic locking and option for cloning upon conflict resolution. (AngularJS, JavaScript, Node.js, PostgreSQL)
- Implemented **Beckon Grid**, an Angular based table configurable via json blobs, with lightning fast rendering that outmatched its rivals. Its configurability (customized cell renderers, cell directives) allowed it to be the central widget across the entire platform. Its fast rendering time (caching rows structure, lazy loading directives) allowed huge amounts of data to be displayed quickly alongside multiple other tables that might be in use on a dashboard. (AngularJS, JavaScript)
- Implemented other key widgets and services to deal with scalability issues such as the throttledRequestService, and requestCachingService. (AngularJS, JavaScript)
- Implemented many layout helpers such as the bnRepositioner to smoothly reposition offscreen rendered items to on screen, and the onTopService to position items at the root of the DOM in the correct position to work around z-index limitations. (AngularJS, JavaScript)

Senior Software Developer **IBM Software Group – Algorithmics / Cognos** **January 2012 – May 2014**

Front End Web Development:

- Developed a highly interactive greenfield SaaS single page web application (SPA) aimed at analyzing counterparty credit risk and credit valuation adjustment (CCR/CVA). (Java, JavaScript, Ext.JS)
- Implemented multiple web based analytic tools including advanced graphing and visualization, grids, searching and report filtering. (JavaScript, Ext.js, D3.js, Highcharts)
- Developed custom cross browser visualizations including scatter plots, vector measure plots, time pickers, and custom domain specific visualizations. (Highcharts)
- Performed extensive graphing optimizations on underlying libraries allowing for larger dataset analysis. (Highcharts)
- Improved data interactivity through computation caching and binary space partitioning, allowing for advanced UI implementations including snapping to important annotated events. (Ext.js, JavaScript)
- Improved data readability through custom SVG based annotations. (Ext.js, JavaScript)
- Experimented with WebGL as a three dimensional visualization technique for spotting local and global minima/maxima, allowing for quick analysis of huge sheets of numbers.(Ext.js, JavaScript, WebGL, three.js)
- Used maintainable practices from the start with proper MVC separation, SASS/CSS for all styling, allowing for rapid prototyping and a completely new look and feel to be quickly experimented with and developed.
- Reversed engineered an entire server side project after losing its sole developer. (Java)
- Increased team productivity by documenting many development, maintenance, and deployment processes on internal wiki.

Software Developer **NexJ Systems** **October 2009 – December 2011**

Client Framework:

- Improved capability and stability of the distributed framework used by all large enterprise customers through extensive bug fixing and feature development. (Java, JavaScript)
- Increased debugging efficiency through clever redefinition of core functions to allow better tracing of asynchronous events, timers, AJAX request/responses, event handlers, and other complex client/server race conditions. (JavaScript)
- Decreased bandwidth requirements up to 90% for classes of stateless requests while simultaneously reducing algorithmic complexity, leading to faster Java, Web, and Blackberry client performance while reducing server load. (JavaScript, Java)

- Discovered the reproduction steps, root cause and fix for a series of long running database query bugs. Further improved product stability through the addition of generic sanity checks to cover other potentially undiscovered race conditions. (Java)
- Proven dedication to product quality and cross-team efficiency as evidenced by extensive bug-tracking searches for issues similar to ones I have solved. It is more efficient to solve similar issues than for another team to redo all the initial investigation.
- Improved product quality by extending the automatic testing framework to support more complex cases involving concurrent sessions across multiple timezones, while significantly expanding JUnit and JavaScript test case coverage. (JavaScript, Java, JUnit)
- Used advanced debugging techniques and tools (WinDbg) to find two causes of an IE9 process crash (double free, corrupted state), and implemented several JavaScript workarounds. (C, JavaScript, WinDbg)
- Authored customer facing documentation explaining the root cause of a complex bug, as well as explanation of the fix. Went the extra mile by describing additional generic coverage. (JavaScript, Java)
- In my spare time I ported the development environment from Windows to Linux, and uncovered some serious bugs before they had a chance to propagate to customers running our server software on Solaris. (Linux, Java)
- Consistently strive for the right fix, which is not necessarily the ‘quick’ fix when time permits. Instead of just fixing the bug at hand, go further where possible to prevent future bugs from being found and filed.

Software Engineer SOMA Networks May 2005 – October 2009

Network Element Web Portal:

- Collected requirements for a cross platform web interface for managing all WiMAX network elements, and devised high and low level designs. (C, JavaScript)
- Implemented a highly interactive portable desktop-like web application using JavaScript, AJAX and XMLHttpRequest, CSS, XHTML, DOM manipulation, and the Ext JS and YUI toolkits. (JavaScript, Ext.js, YUI)
- Web Client includes device and subscriber monitoring and discovery, removable hardware detection, X.731 alarm management, and logging. (C)
- Implemented all server side functionality using Lua, Kepler's Xavante, CGI with proprietary C modules, and database linkages. (Lua, C)
- Authorization and authentication functionality provided through PAM linkages. (Linux)

WiMAX SNMPv3 Embedded Monitoring System:

- Created an AgentX sub-agent to provide vital Mobile Subscriber, Networking and Radio statistics, using NetSNMP and eXtremeDB in an embedded Linux/VxWorks environment. The sub-agent is crucial to the functioning of centralized EMS and NMS systems (C)
- Added advanced multi-threading support to NetSNMP. (C)
- Drastically improved NetSNMP scalability by inventing a proprietary *$O(n \log n)$* in-memory storage system to replace the existing *$O(n^2)$* system. This allowed millions of rows to be parsed in tight time constraints instead of thousands.
- Tracked down and removed a large number of NetSNMP memory leaks. (C)
- Added SNMPv3 context-Notification support. (C)

VoIP Network Management System:

- Designed, documented and implemented the SNMPStats module for the OpenSER, OpenSIPS, and Kamilio SIP systems. The widely adopted module provides SNMP based monitoring and statistics gathering, allowing for the collection of important business and operational statistics. (C)

Common Event Interface Communication Engine:

- Coded a standardized interface communication engine for interprocess communication of complex objects across arbitrary networks. (C)
- Increased capacity 30 fold by designing a more efficient serialization system that minimized memory allocation and fragmentation. (C)

Solaris Core Server Scalability Metrics:

- Improved manageability of core server systems by adding SNMP based scalability metrics, allowing for fine tuning and testing of server parameters. (Java)
- Cut down development time by two thirds through clever use of Java reflection. (Java)

Software Engineer – Driver Development (PEY) **AMD (Formerly ATI Technologies)** **May 2003 - September 2004**

HDTV Driver Development:

- Paved the way for using HDTV's as part of a desktop system by conceiving, researching, designing and implementing customized mode support for HDTV drivers. (C/ASM).
- Implemented testing infrastructures for HDTV drivers and Display Abstraction Layer features. (C++).
- Uncovered and removed an OverDrive BSOD bug involving race conditions between thermal interrupts, 2D Drivers, and PLL clock settings via kernel debugging with WinDbg. (C/ASM)

NSERC Research Assistant – Dynamic Graphics Project (DGP) **University of Toronto** **May 2002 – January 2003**

StretchMat Prototyping:

- Key player in the research and design of innovative human-computer interaction devices, resulting in several bend, twist, and stretch sensitive 3D aware surface prototypes. (C++)
- Innovated algorithms to map the StretchMat's fiber-optic spiral sensor network to virtual 3D surfaces and objects. (C++)
- Researched and experimented with 2D and 3D spatial algorithms (including KD Trees) and innovated applications to support modeless interaction with a 3D scene through gesture recognition. (C++)

Education

Honors B.Sc. Computer Science (Software Engineering Specialist) **University of Toronto** **Sept 2000 – May 2005 (With 16 month PEY Co-op)**

- Studied Web Development (web based applications, e-commerce), User Interface Design (Human Computer Interaction), Computer Graphics and Visualization (anti-aliasing, image manipulation, video compression, 3D Modeling), Artificial Intelligence (Bayesian Learning, Natural Language Understanding, Neural Networks), Operating Systems, Embedded Systems, Linux Device Drivers, Time Complexity, Software Engineering Practices, and Data Structures.
- Awarded NSERC Research Scholarship after second year for the Dynamic Graphics Project Research Group

Summary of Qualifications

Software Development Lifecycle:

- Over 21 years of professional software development as a full stack engineer.
- Cofounder and principal engineer of Powder and Y Combinator Alumni, with $5MM raised.
- Developed multiple startup products from scratch as a founding engineer, bringing multiple concepts to sellable products, while also experienced working with larger teams.
- Consistently architect elegant and easily deployable designs, which translate to efficient and maintainable coded solutions.
- Efficiently track down and correct a wide range of complicated bugs consisting of race conditions, memory leaks, and stack corruption in other existing software.
- Expert in working with large foreign code bases as demonstrated through extensive open source contributions.

Technological and Domain Knowledge:

- Worked extensively with React, Angular, JavaScript/ES6, NodeJS, MongoDB, PostgreSQL, Python, Django, FastAPI, Java, C/C++, web3, Ethers.js (Cryptocurrency), JSON, XML, CSS, SQL.
- Built multiple cloud computing based platforms on AWS and DigitalOcean using microservices.

- Integrated multiple software projects with LLMs for groundbreaking AI delivery.
- Avid reader of arxiv.org research papers, following the latest developments in AI and building LLMs.
- Familiar with network diagnostic software such as tcpdump, ethereal/wireshark, traceroute, etc.
- Worked extensively with Linux (16+ years), MacOS and Windows.

Communication and Interpersonal:

- Superior writing skills as demonstrated through extensive developer and end-user documentation.
- Strong team player with well developed leadership skills.
- Proven ability to work and learn independently, with minimal supervision.

References Available Upon Request