

# Jon Howell

6827 18th Ave NE  
Seattle, WA 98115

jonh.cv@jonh.net

January 2024

## RESEARCH STATEMENT

My contributions focus on correctness, consistency, distributed systems, and security.

In the last decade, my primary focus is bringing verification to practical impact. Just as engineers today invest in strong types and ownership types to obviate big classes of runtime bugs, I aim to make verification a practical tool to statically eliminate logic bugs. Today's automated verification tools are tantalizingly close to practical. My modus operandi is to push these tools hard on big systems, finding which features are the bottleneck, and creating a feedback loop that refines the tools.

I contribute to this space by:

- Building a high-performance storage system that pushes verification and its software development process beyond previous limits.
- Bringing knowledge back from that process to improve heap reasoning techniques, concurrency reasoning schemes, debugging strategies, and inference techniques. The Verus verification language grew out of our storage system project.
- Teaching verification software engineering techniques to develop the methodology and understand how it is utilized by working engineers. I have co-developed a course taught at VMware and Michigan which have led two VMware teams to employ protocol verification.

## PROFESSIONAL BACKGROUND

**Principal Researcher**, *VMware Research*, Bellevue, WA.

August 2018 - Present

**Staff Engineer**, *Google*, Kirkland, WA.

May 2015 - August 2018

**Researcher**, *Microsoft Research*, Redmond, WA.

August 2001 - April 2015

**Software Engineer**, *Consystant Systems*, Kirkland, WA.

July 2000 - August 2001

## EDUCATION

**Dartmouth College**, Hanover, NH.

September 1995 - May 2000

Ph.D. in Computer Science

Dissertation: *Naming and Sharing Resources Across Administrative Boundaries*.

Advisor: David Kotz

**University of Puget Sound**, Tacoma, WA.

September 1990 - May 1994

Summa Cum Laude with a BS in Computer Science, BS in Mathematics

## HONORS

IEEE Symposium on Security and Privacy 2023 (Oakland) **Test of Time** Award for *Pinocchio: Nearly Practical Verifiable Computation*, S&P 2013.

ASPLOS 2022 **Influential Paper** Award

for *Rethinking the Library OS from the Top Down*, ASPLOS 2011.

**Distinguished Paper** Award for *Linear Types for Large-Scale Systems Verification*, OOPSLA, 2022.

Research Highlight for *IronFleet*, Communications of the ACM, 2017.

Research Highlight for *Pinocchio*, Communications of the ACM, 2016.

**Best Paper** Award for *Pinocchio*, IEEE Symposium on Security and Privacy (Oakland), 2013.

**Best Paper** Award for *Embassies*, USENIX Symposium on Networked Systems Design & Impl. (NSDI), 2013.

**World Record**, Gray Minute Sort Benchmark, for *Flat Datacenter Storage*, 2012.

USENIX Graduate Fellowship, 1997 - 2000.

## JOURNALS

*IronFleet: Proving Practical Distributed Systems Correct.*

Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R. Lorch, Bryan Parno, Michael L. Roberts, Srinath Setty, and Brian Zill.

Communications of the ACM (**CACM**), July, 2017.

**Research Highlight.**

*Pinocchio: Nearly Practical Verifiable Computation.*

Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova.

Communications of the ACM (**CACM**), February, 2016.

**Research Highlight.**

## REFEREED

## CONFERENCES

*Verus: Verifying Rust Programs using Linear Ghost Types.*

Andrea Lattuada, Travis Hance, Chanhee Cho, Matthias Brun, Isitha Subasinghe, Yi Zhou, Jon Howell, Bryan Parno, and Chris Hawblitzel.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (**OOPSLA**), October 2023.

*Leaf: Modularity for Temporary Sharing in Separation Logic.*

Travis Hance, Jon Howell, Oded Padon, and Bryan Parno.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (**OOPSLA**), October 2023.

*Counterexample Driven Quantifier Instantiations with Applications to Distributed Protocols.*

Orr Tamir, Marcelo Taube, Kenneth L. McMillan, Sharon Shoham, Jon Howell, Guy Gueta, Mooly Sagiv.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (**OOPSLA**), October 2023.

*Sharding the State Machine: Automated Modular Reasoning for Complex Concurrent Systems.*

Travis Hance, Yi Zhou, Andrea Lattuada, Reto Achermann, Alex Conway, Ryan Stutsman, Gerd Zellweger, Chris Hawblitzel, Jon Howell, and Bryan Parno.

USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), July 2023.

*Linear Types for Large-Scale Systems Verification.*

Jialin Li, Andrea Lattuada, Yi Zhou, Jonathan Cameron, Jon Howell, Bryan Parno, and Chris Hawblitzel.

ACM Conference on Object-Oriented Programming Systems, Languages, and Applications (**OOPSLA**), December 2022.

**Distinguished Paper Award.**

*Storage Systems are Distributed Systems (So Verify Them That Way!).*

Travis Hance, Andrea Lattuada, Chris Hawblitzel, Jon Howell, Rob Johnson, and Bryan Parno.

USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), November 2020.

*Slicer: Auto-Sharding for Datacenter Applications.*

Atul Adya, Daniel Myers, Jon Howell, Jeremy Elson, Colin Meek, Vishesh Khemani, Stefan Fulger, Pan Gu, Lakshminath Bhuvanagiri, Jason Hunter, Roberto Peon, Larry Kai, Alexander Shraer, Arif Merchant, and Kfir Lev-Ari.

USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), November 2016.

CONFERENCES  
CONTINUED

*IronFleet: Proving Practical Distributed Systems Correct.*

Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R. Lorch, Bryan Parno, Michael L. Roberts, Srinath Setty, and Brian Zill.  
ACM Symposium on Operating Systems Principles (**SOSP**), October 2015.

*Geppetto: Versatile Verifiable Computation.*

Craig Costello, Cedric Fournet, Jon Howell, Markulf Kohlweiss, Benjamin Kreuter, Michael Naehrig, Bryan Parno, and Samee Zahur.  
IEEE Symposium on Security and Privacy (**Oakland**), May 2015.

*Ironclad Apps: End-to-End Security via Automated Full-System Verification.*

Chris Hawblitzel, Jon Howell, Jacob R. Lorch, Arjun Narayan, Bryan Parno, Danfeng Zhang, and Brian Zill.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), October 2014.

*Missive: Fast Application Launch From an Untrusted Buffer Cache.*

Jon Howell, Jeremy Elson, Bryan Parno, and John R. Douceur.  
USENIX Annual Technical Conference (**ATC**), June 2014.

*How to Run POSIX Apps in a Minimal Picoprocess.*

Jon Howell, Bryan Parno, and John R. Douceur.  
USENIX Annual Technical Conference (**ATC**), June 2013.

*Pinocchio: Nearly Practical Verifiable Computation.*

Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova.  
IEEE Symposium on Security and Privacy (**Oakland**), May 2013.  
**Best Paper Award.**  
**Test of Time Award, 2023.**

*Embassies: Radically Refactoring the Web.*

Jon Howell, Bryan Parno, and John R. Douceur.  
USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), April 2013.  
**Best Paper Award.**

*Flat Datacenter Storage.*

Edmund B. Nightingale, Jeremy Elson, Jinliang Fan, Owen Hofmann, Jon Howell, and Yutaka Suzue.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), October 2012.

*Rethinking the Library OS from the Top Down.*

Donald E. Porter, Silas Boyd-Wickizer, Jon Howell, Reuben Olinsky, and Galen Hunt.  
Architectural Support for Programming Languages and Operating Systems (**ASPLOS**), March 2011.  
**Influential Paper Award 2022**

*The Utility Coprocessor: Massively Parallel Computation from the Coffee Shop.*

John R. Douceur, Jeremy Elson, Jon Howell, and Jacob R. Lorch.  
USENIX Annual Technical Conference (**ATC**), June 2010.

*What You See is What They Get: Protecting users from unwanted use of microphones, cameras, and other sensors.*

Jon Howell and Stuart Schechter.  
Web 2.0 Security and Privacy, May 2010.

*Crom: Faster Web Browsing Using Speculative Execution.*

James Mickens, Jeremy Elson, Jon Howell, and Jay Lorch.  
USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), April 2010.

CONFERENCES  
CONTINUED

- Mugshot: Deterministic Capture and Replay for JavaScript Applications.*  
James Mickens, Jeremy Elson, and Jon Howell.  
USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), April 2010.
- Leveraging Legacy Code to Deploy Desktop Applications on the Web.*  
John R. Douceur, Jeremy Elson, Jon Howell, and Jacob R. Lorch.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), December 2008.
- Low-Cost Orthographic Imagery.*  
Peter Pesti, Jeremy Elson, Jon Howell, Drew Steedly, and Matt Uyttendaele.  
ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (**ACM GIS**), November 2008.
- Handling Flash Crowds from your Garage.*  
Jeremy Elson and Jon Howell.  
USENIX Annual Technical Conference (**ATC**), June 2008.
- "Do I Live in a Flood Basin?": Synthesizing Ten Thousand Maps.*  
Miguel Elias, Jeremy Elson, Danyel Fisher, and Jon Howell.  
Human Factors in Computing Systems (**CHI**), April 2008.
- Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization.*  
Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul.  
Computer and Communications Security (**CCS**), October 2007.
- Protection and Communication Abstractions for Web Browsers in MashupOS.*  
Helen Wang, Xiaofeng Fan, Jon Howell, and Collin Jackson.  
ACM Symposium on Operating Systems Principles (**SOSP**), 2007.
- Distributed Directory Service in the FARSITE File System.*  
John R. Douceur and Jon Howell.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), November 2006.
- The SMART Way to Migrate Replicated Stateful Services.*  
Jacob R. Lorch, Atul Adya, William J. Bolosky, Ronnie Chaiken, John R. Douceur, and Jon Howell.  
European Conference on Computer Systems (**EuroSys**), April 2006.
- FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment.*  
Atul Adya, William J. Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R. Douceur, Jon Howell, Jacob R. Lorch, Marvin Theimer, and Roger P. Wattenhofer.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), December 2002.
- Cooperative Task Management without Manual Stack Management.*  
Atul Adya, Jon Howell, Marvin Theimer, William J. Bolosky, and John R. Douceur.  
USENIX Annual Technical Conference (**ATC**), June 2002.
- End-to-End Authorization.*  
Jon Howell and David Kotz.  
USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), October 2000.
- A Formal Semantics for SPKI.*  
Jon Howell and David Kotz.  
European Symposium on Research in Computer Security (**ESORICS**), October 2000.

- CONFERENCES  
CONTINUED** *Practical Mobile Robot Self-Localization.*  
Jon Howell and Bruce Randall Donald.  
International Conference on Robotics and Automation (**ICRA**), April 2000.
- Experiments with Desktop Mobile Manipulators.*  
Matthew T. Mason, Dinesh K. Pai, Daniela Rus, Jon Howell, Lee R. Taylor, and Michael A. Erdmann.  
International Symposium on Experimental Robotics (**ISER**), March 1999.
- INVITED  
CHAPTERS** *The FARSITE Project: a Retrospective.*  
William J. Bolosky, John R. Douceur, and Jon Howell.  
SIGOPS Operating Systems Review (**OSR**), April 2007.
- MapCruncher: Integrating the World's Geographic Information.*  
Jeremy Elson, Jon Howell, and John R. Douceur.  
SIGOPS Operating Systems Review (**OSR**), April 2007.
- WORKSHOPS** *Domesticating Automation.*  
Pranav Srinivasan, Oded Padon, Jon Howell, and Andrea Lattuada.  
**Dafny** workshop at POPL, January 2024.
- Beyond Isolation: OS Verification as a Foundation for Correct Applications.*  
Matthias Brun, Reto Achermann, Tej Chajed, Jon Howell, Gerd Zellweger, and Andrea Lattuada.  
Hot Topics in Operating Systems (**HotOS**), June 2023.
- An Incremental Path Towards a Safer OS Kernel.*  
Jialin Li, Samantha Miller, Danyang Zhuo, Ang Chen, Jon Howell, and Thomas Anderson.  
Hot Topics in Operating Systems (**HotOS**), June 2021.
- The Web Interface Should Be Radically Refactored.*  
John R. Douceur, Jon Howell, Bryan Parno, Michael Walfish, and Xi Xiong.  
Hot Topics in Networks (**HotNets**), November 2011.
- Refactoring Human Roles Solves Systems Problems.*  
Jeremy Elson and Jon Howell.  
Hot Topics in Cloud Computing (**HotCloud**), June 2009.
- MashupOS: Operating System Abstractions for Client Mashups.*  
Jon Howell, Collin Jackson, Helen Wang, and Xiaofeng Fan.  
Hot Topics in Operating Systems (**HotOS**), May 2007.
- Byzantine Fault Isolation in the FARSITE Distributed File System.*  
John R. Douceur and Jon Howell.  
International Workshop on Peer-to-Peer Systems (**IPTPS**), February 2006.
- Hey, You Got Your Compiler In My Operating System!*  
Jon Howell and Mark Montague.  
Hot Topics in Operating Systems (**HotOS**), March 1999.
- Straightforward Java persistence through checkpointing.*  
Jon Howell.  
International Workshop on Persistence and Java, September 1998.

**TECHNICAL  
REPORTS**

*Living Dangerously: A Survey of Software Download Practices.*  
Jon Howell, Galen Hunt, David Molnar, and Donald E. Porter.  
Technical Report MSR-TR-2010-51, May 2010.

*Replicated Virtual Machines.*  
John Douceur and Jon Howell.  
Technical Report MSR-TR-2005-119, September 2005.

*Black Box Leases.*  
John Douceur and Jon Howell.  
Technical Report MSR-TR-2005-120, September 2005.

*Replicated Virtual Machines.*  
John Douceur and Jon Howell.  
Technical Report MSR-TR-2005-119, September 2005.

*Load Management in a Large-Scale Decentralized File System.*  
Atul Adya, William J. Bolosky, Ronnie Chaiken, John R. Douceur, Jon Howell, and Jacob Lorch.  
Technical Report MSR-TR-2004-60, July 2004.

*Correctness of Paxos with Replica-Set-Specific Views.*  
Jon Howell, Jacob Lorch, and John Douceur.  
Technical Report MSR-TR-2004-45, June 2004.

*Scalable Byzantine-Fault-Quantifying Clock Synchronization.*  
John Douceur and Jon Howell.  
Technical Report MSR-TR-2003-67, October 2003.

*Landmarks for Absolute Localization.*  
Jon Howell and Keith Kotay.  
Technical Report TR 2000-364, Dartmouth College, 2000.

**PROFESSIONAL  
ACTIVITIES**

**Steering Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), 2020 - Present.

**Steering Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2017 - Present.

**Program Committee**, ACM Symposium on Operating Systems Principles (**SOSP**). 2024.

**Program Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). 2024.

**Program Committee**, Hot Topics in Operating Systems (**HotOS**). 2023.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2023.

**Program Committee**, USENIX Annual Technical Conference (**ATC**). 2022.

**Program Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). 2021.

**Program Committee**, Hot Topics in Operating Systems (**HotOS**). 2021.

**Program Committee**, USENIX Workshop on Hot Topics in Cloud Computing (**HotCloud**). 2020.

**PC Co-Chair**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**), 2020.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2019.

**Program Committee**, European Conference on Computer Systems (**EuroSys**), 2019.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2018.

**Program Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). 2018.

**PC Co-Chair**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2017.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2017.

**Program Committee**, ACM Symposium on Operating Systems Principles (**SOSP**). 2017.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2016.

**Program Committee**, USENIX Annual Technical Conference (**ATC**). 2016.

**Program Committee**, USENIX Symposium on Networked Systems Design and Implementation (**NSDI**), 2015.

**Program Committee**, USENIX Annual Technical Conference (**ATC**). 2014.

**Program Committee**, ACM Symposium on Operating Systems Principles (**SOSP**). 2013.

**Program Committee**, USENIX Annual Technical Conference (**ATC**). 2013.

**Track Chair**, International Conference on Distributed Computing Systems (**ICDCS**). 2013.

**Program Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). 2012.

**Program Committee**, USENIX Annual Technical Conference (**ATC**). 2012.

**Program Committee**, Web Apps. 2011.

**Program Committee**, USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). 2010.

**Program Committee**, Web Apps. 2010.

**Program Committee**, Workshop on Programming Support Innovations for Emerging Distributed Applications (**PSI ETA**). 2010.

**PROFESSIONAL ACTIVITIES CONTINUED**     **Program Committee**, ACM Workshop on Large-Scale System and Application Performance (LSAP). 2009.  
**Program Committee**, International Conference on Distributed Computing Systems (ICDCS). 2009.  
**Program Committee**, USENIX Conference on File and Storage Technologies (FAST). 2005.

**MENTORING & TEACHING**     **Courses**  
Protocol Verification, *VMware engineering attendance*.  
May 2023, with Tej Chajed.  
Verification Summer School, *Hosted at VMware, world attendance*.  
August 2021, with Manos Kapritsos.  
Verification Summer School, *Hosted at VMware, world attendance*.  
August 2020, with Manos Kapritsos.  
Distributed Systems, *University of Washington*.  
Spring 2019, with Jay Lorch.

**Thesis Committees**  
Xudong Sun, *UIUC*, underway.  
Yi Zhou, *Carnegie Mellon*, underway.  
Jialin Li, *University of Washington*, underway. Coadvisor.  
Luke Nelson, *University of Washington*, Defended September, 2023.  
Shih-Wei Li, *Columbia University*. Defended March, 2021.  
Raymond Cheng, *University of Washington*. Defended August, 2017.  
Sangmin Lee, *University of Texas, Austin*. Defended August, 2014.

**PATENTS**

<i>End-to-End Security via Secure Hardware Running Verified Software.</i>	#9,363,087 – 2016
<i>Reading and writing during cluster growth phase.</i>	#8,447,833 – 2012
<i>Embedded wireless cloud connector.</i>	#8,682,957 – 2012
<i>Application compatibility shims for minimal client computers.</i>	#8,903,705 – 2010
<i>Secure application interoperation via user interface gestures.</i>	#8,335,991 – 2010
<i>Collection ordering for replicated state machines.</i>	#8,135,987 – 2010
<i>Capturing web-based scenarios.</i>	#8,407,321 – 2010
<i>Bandwidth-proportioned datacenters.</i>	#8,438,244 – 2010
<i>Memory management and recovery for datacenters.</i>	#8,181,061 – 2010
<i>Locator table and client library for datacenters.</i>	#8,533,299 – 2010
<i>Network application performance enhancement using speculative execution.</i>	#8,140,646 – 2009
<i>Leveraging remote server pools for client applications.</i>	#8,549,106 – 2009
<i>Splitting file types within partitioned images.</i>	#8,139,872 – 2008
<i>Defeating cache resistant domain name systems.</i>	#7,865,618 – 2008
<i>Domain name cache control system generating series of varying nonce-bearing domain names based on a function of time.</i>	#7,958,261 – 2008
<i>Human performance in human interactive proofs using partial credit.</i>	#8,209,741 – 2007
<i>Interest aligned manual image categorization for human interactive proofs.</i>	#8,104,070 – 2007
<i>Tiled packaging of vector image data.</i>	#7,925,100 – 2007
<i>Adding custom content to mapping applications.</i>	#8,749,585 – 2007
<i>Transforming offline maps into interactive online maps.</i>	#8,368,695 – 2007
<i>Non-mutating tree-structured file identifiers.</i>	#7,574,457 – 2006
<i>Managing states with delta pager.</i>	#7,430,553 – 2005
<i>Scalable leases.</i>	#7,650,514 – 2005
<i>Transaction and task scheduler.</i>	#7,716,249 – 2005
<i>Automatic commutativity detection for generalized paxos.</i>	#8,046,413 – 2005
<i>Method and system for protecting the consistency of information in a distributed file system.</i>	#7,783,664 – 2004
<i>Efficient changing of replica sets in distributed fault-tolerant computing system.</i>	#7,334,154 – 2004
<i>Byzantine fault quantifying clock synchronization.</i>	#7,454,521 – 2003