RH RQ

Bringing great research ideas into open source communities

Mo Duffy and Rudolph Pienaar

ChRIS five years later: leveling the playing field for advanced analytics and AI

+

Ecological forecasting: from supercomputer to cloud

Function as a Service in the cloud continuum

Team threat hunting on a container platform: Kestrel as a Service

Red Hat
Research Quarterly
Volume 6:1 | May 2024 | ISSN 2691-5278

MUNI
FI

Masaryk University
Faculty of Informatics

Your **research,**

projects

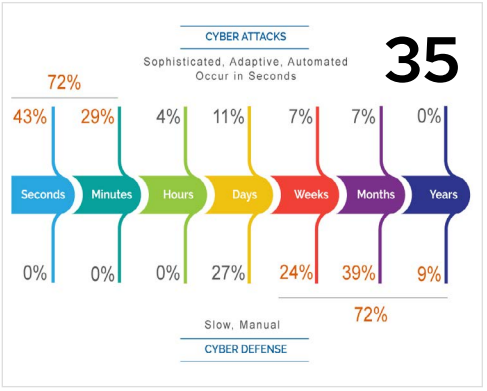
and **education** partner.

FI.MUNI.CZ

RESEARCH QUARTERLY

VOLUME 6:1

Table of Contents



Departments

Features

Red Hat

RESEARCH QUARTERLY

VOLUME 6:1

From the director

Beyond the aha! moment: research develops solutions for lasting impact

by Heidi Picher Dempsey

About the Author
Heidi Picher Dempsey is the US Research Director for Red Hat. She seeks and cultivates research and open source projects with academic and commercial partners in operating systems, hybrid clouds, performance optimization, networking, security, and distributed system operations.

When Alexander Graham Bell made the first telephone call in his Boston laboratory in 1876, saying to his assistant "Mr. Watson, come here—I want to see you," he created our popular image of the inventor's aha! moment. What that image doesn't show is that this was just one moment in a research, development, and deployment effort spanning many years and a much larger group of collaborators bringing telephony into nearly everyone's home and eventually into everyone's hand as well.

Bell and Watson differed in their journal accounts of this moment, but there is no question this event could never have happened without their partnership. It is also interesting that Bell's original work was as an elocutionist, not primarily as an inventor, and his early goal was to improve the quality of life and opportunities for people with hearing impairments. This issue of *Red Hat Research Quarterly* highlights R&D projects that, although they engendered their own aha! moments, primarily create real value by bringing together very different components into systems that make it possible for people far removed from the daily development of technology to do important work in their own worlds.

This issue also highlights the importance of building workflows, not just clever point solutions. The ChRIS project, which began as a partnership between a doctor and a computer scientist investigating what they could do together to make it easier for clinicians to evaluate and diagnose patients, grew into a bigger collaboration with Red Hat, the Mass Open Cloud, Boston Children's Hospital, and the open source developer community over several years. In conversation with Orran Kreiger from the MOC, Máirín Duffy, who led the effort to develop a thoughtful and efficient user experience with ChRIS tools, and Rudolph Pienaar, the principal investigator for ChRIS research, provide fascinating insights into why they work on ChRIS, how AI will affect what's coming next, and how to break down silos between hospitals and developers with an open cloud platform.

The PEcAn project provides another great example of how ecologists and developers can come together to create a new workflow. Shared AI workbenches in PEcAn will allow scientists to respond to extreme weather events and evolving environmental changes by analyzing large troves of existing environmental data in new ways. Red Hat Principal Engineer Chris Tate teamed up with BU Professor Michael Dietz to develop a cloud architecture that would be accessible to ecologists worldwide, leveraging existing ecological models and data from Dietz's previous projects.

Red Hat

Tate built on existing OpenShift software and worked with Dietz's team to develop an Advanced Messaging Queuing Protocol (AMQP) that receives messages to run forecasts and a scalable PostgreSQL database with built-in geolocation features that are all running currently in the Mass Open Cloud. In his RHRQ article, Tate relates an interesting personal history of how the team worked together to build PEcAn and how they are working together to share their solutions with other ecologists.

Getting collaborators in a single city, or a single scientific area, to bring components to build a new larger system is challenging, as these first two articles explain. In Europe, the challenge reaches the international level. This issue's retrospective on the PHYSICS project, (oPtimized Hybrid Space-Time Servlce Continuum in faaS), which has a name straight out of Star Trek, tells the story of many international collaborators who worked together for several years to take the idea of serverless computing where no person had gone before. The team built a novel architecture to enable easier software deployments across diverse clusters in complex real-world environments. In addition to substantial open source software contributions (Kepler, Submariner, Knative) the team demonstrated how to use the architecture to build systems to benefit e-health, smart agriculture, and smart manufacturing communities.

Enjoy these fascinating stories of how individual technical partnerships can grow into solutions that actually help us live long and prosper.

THERE ARE MANY UNIVERSITIES IN MASSACHUSETTS, BUT ONLY ONE FLAGSHIP *FOR* MASSACHUSETTS.

As the Commonwealth's flagship public research university, UMass Amherst is committed to pursuing progress for our great state in computer science, technology, engineering, and more. Learn why we've soared to #26 in *U.S. News & World Report* rankings of top-tier public universities and find your degree.

Learn more at **umass.edu**

University *of* Massachusetts Amherst

RESEARCH QUARTERLY

VOLUME 6:1

Red Hat

News

How the university OSPO approach is taking shape at the University of California and beyond

Universities are looking at open source program offices to make the most of open source research on campus. UC Santa Cruz has an update on their progress.

by Stephanie Lieggi, Emily Lovell, and James Davis

In 2022, the University of California, Santa Cruz became one of the first universities to build an academic open source program office (OSPO) on their campus (see "Building a university OSPO: bolstering academic research through open source," RHRQ Feb. 2022).

The OSPO's goals are to create an infrastructure for open source ecosystems on campus, improve the talent pipelines through teaching and mentorship, and expand the reach and impact of research coming from the university. Two years on, this approach is taking root not just at UCSC but throughout the UC system, including a current effort to create a UC-wide network of OSPOs.

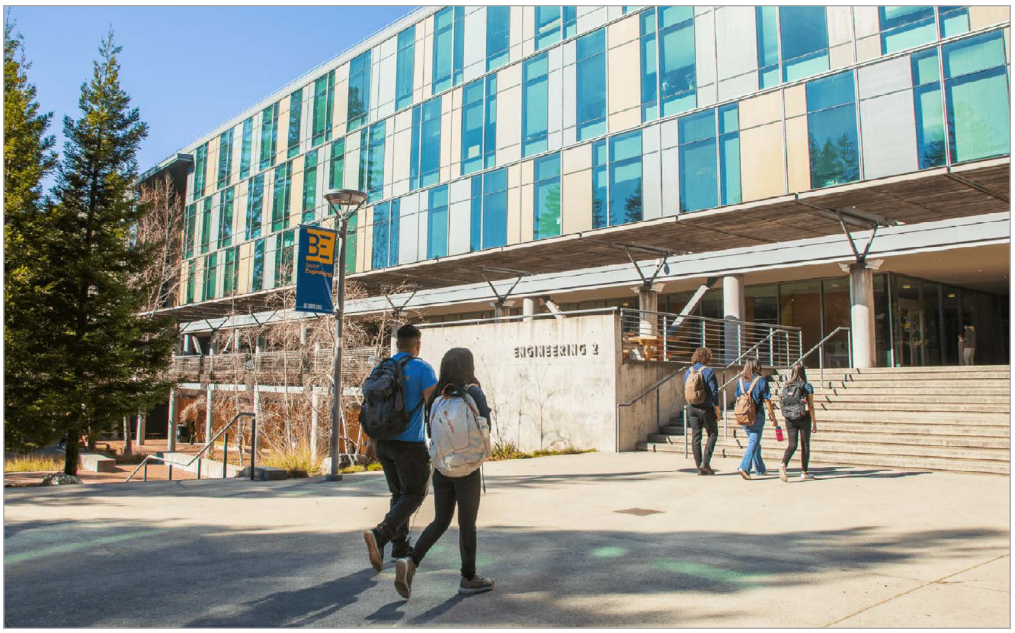
GROWING INFLUENCE

While academics are sometimes unfamiliar with the role of open source in research, leadership on a growing number of campuses is recognizing that promoting an open approach has a positive impact on their school's research standing.

Engaging in open source also enables them to tap into a more diverse set of funders. This recognition is evident in the activities of groups such as the Higher Education Leadership Initiative for Open Scholarship (HELIOS). HELIOS brings together more than 60 top-tier universities to coordinate efforts that align higher education practices with open scholarship values. Launched in 2022, this community of practice aims to elevate open practices and develop campus implementation of open research strategies. While HELIOS does not specifically call for the development of university OSPOs, the results of their work promote infrastructure and policies that run parallel to burgeoning OSPOs.[1]

The number of academic OSPOs is small but growing steadily. Support from the Sloan Foundation—which has funded 12 OSPO pilot

[1] See Helios, *Guide to supporting open scholarship for university presidents and provosts*, Helios Open.



The Baskin Engineering complex is home to CROSS and the UCSC OSPO.

projects to date[2]—and infrastructure-focused programs such as the National Science Foundation's Pathways to Enable Open Source Ecosystems (POSE) has empowered academic OSPOs to help researchers and scientists build on the work of others, make academic research more replicable, and further universities' education and public service missions.

In the process, these OSPOs are building on one another's successes through platforms that allow for experience sharing, including the Sustain Academic Working Group, which facilitates open source focused collaborations in academic institutions; the metrics-focused CHAOSS (Community Health Analytics in Open Source Software) University Open Source Working Group; and the Community for University & Research Institute OSPOs (CURIOSS), which facilitates collaborations between OSPOs in universities and research institutions. While focusing significant efforts on building local campus infrastructure, the current cohort of academic OSPOs has also used these platforms to leverage the work of others and enable newcomers to benefit from lessons learned.

THE UC SANTA CRUZ OPEN APPROACH

UCSC has long recognized the value of open source and open science. As part of the publicly funded Human Genome Project, UCSC researchers were the first to publish the human genome in 2000 as an open source artifact, beating a privately funded group intent on patenting the discovery. As a result, scientists worldwide can freely tap this knowledge and advance the understanding of human biology, evolution, and disease. The success of this project, and the resulting Human Genome Browser, illustrated the power of openness in increasing the impact of university research and bringing significant value to campus research.

Support for openness on the Santa Cruz campus drove the creation of the Center for Research in Open Source Software (CROSS) and the development of the UCSC OSPO. The origins of CROSS lay in the Ceph open source project created by UCSC PhD student Sage Weil. Weil continued to develop Ceph and, in 2014, sold his start-up to Red Hat, gifting $2 million of the proceeds to UCSC professor and Ceph co-creator Carlos Maltzahn. Maltzahn founded CROSS in 2015 and began partnering with industry to provide researchers the opportunity to build communities and ecosystems around their open source projects. By 2021, CROSS raised an additional $2.6 million through industry membership and supported the work of graduate students and postdoctoral scholars building new and innovative open source approaches through their research.

In 2020, CROSS became an early participant in groups like OSPO++, which promoted the need for OSPOs

[2] Apart from UCSC, the originally funded campuses include Carnegie Mellon University, Rochester Institute of Technology, John Hopkins University, St Louis University, and the University of Vermont. In 2023, six more universities were funded: Georgia Tech, George Washington University, Stanford University, Syracuse University, University of Texas in Austin, and the University of Wisconsin-Madison.

outside of industry, particularly in academia and the public sector. The CROSS team saw significant gaps in their ability to leverage open source in the university setting that an OSPO approach could fill, including insufficient training of students in open source, lack of coherence in technology transfer approaches, lost opportunities for university-industry partnerships around open source products, and an ongoing failure of replicability of academic research. While CROSS, as a research center in the Baskin School of Engineering, could bridge some of those gaps, a true campus-wide, multidisciplinary approach was needed to leverage the potential power of open source fully. Building an OSPO at UCSC provided a clear opportunity to expand activities that highlight the value of open source to the university research enterprise.

EMPOWERING THE NEXT GENERATION

The UCSC OSPO undertakes efforts to promote open source research in parallel with training and mentorship programs that began under CROSS. In 2018, CROSS became a mentor organization for the Google Summer of Code (GSoC). Through this program, UCSC's GSoC mentors—typically PhD students or postdocs developing open source efforts around their own research projects—were able to build community and gain maintainer experience. Out of this positive experience, the Open Source Research Experience (OSRE) program was established.

This program—which still leverages participation in programs like GSoC—now supports mentors throughout the UC system, allowing researchers to build a diverse community of contributors and provide hands-on experience to open source newcomers. With the support of the NSF, the OSRE also established the Summer of Reproducibility, following the GSoC model, to focus on research projects that produce and use reproducibility artifacts. In 2024, the OSRE—now formally under the UCSC OSPO—supported 37 students from 11 countries working with 40 mentors, completing summer projects that contributed to open source research efforts and practical reproducibility.

> OSPOs are uniquely situated to meet the challenge of strengthening and diversifying the open source workforce.

Strengthening and diversifying the open source workforce is a critical issue for open source communities generally, and academic OSPOs are uniquely situated to meet this ongoing challenge. In 2023, in collaboration with historically Black university Norfolk State University, the UCSC OSPO and Baskin Engineering Inclusive Excellence Hub launched the OSRE Contributor Catalyst program. This pilot was an eight-week hybrid program in which four NSU students learned to productively engage in open source. Participating NSU students spent about four weeks learning about open source, followed by four weeks remotely contributing to an open source project, with half of this time spent together on campus at UCSC and half spent collaborating remotely. The program provided a mix of mentorship, specialized instruction, and community building that supported participants in considering both industry and career pathways in the field. A unique feature of this program was its cohort structure, meaning that students lived, learned, and worked closely together for the duration, supporting one another throughout. Given the success of this pilot offering, the UCSC OSPO plans to grow the program to include multiple HBCU partners (and many more students) over the next few years.

BUILDING THE UC NETWORK

With the help of a recent grant from the Sloan Foundation, the UCSC OSPO has partnered with five other UC campuses—Berkeley, Davis, Los Angeles, San Diego, and Santa Barbara—to build a network of local OSPOs throughout the UC system. The project aims to create systemwide communities of practice that can leverage the experience of individual campuses, yielding a robust network of open source practitioners. This multicampus collaboration can also be a strong catalyst for increasing the understanding of the current open source landscape in the UC system.

While the UCSC OSPO has seen some success in highlighting the value of open source to the UCSC

About the Author

Stephanie Lieggi is the Executive Director at CROSS and the University of California, Santa Cruz OSPO, supporting the work of university-based open source projects and communities.

About the Author

Emily Lovell is a Postdoctoral Fellow in the University of California, Santa Cruz OSPO, where she works to support diverse, equitable, and inclusive student engagement in open source.

About the Author

James Davis is the Director of CROSS and Professor of Computer Science and Engineering at University of California, Santa Cruz. His research including both traditional computer science and technology applications to social issues has resulted in over 100 peer-reviewed publications.

research communities, the current methodology for discovering the true scope of the on-campus open source landscape is insufficient and often misses potentially innovative research happening on campus. This lack of understanding limits the ability of UCSC, as well as other academic OSPOs, to effectively connect the university research enterprise with relevant open source communities and potential sponsors. In an initial exercise, UC San Diego researchers discovered at least 32,000 repositories on GitHub connected to their campus, and a similar scan of GitHub revealed at least 13,000 repos connected to UCSC, revealing that open source activity on campus was well beyond what had previously been identified.

As part of the effort to build a UC OSPO network, UCSC and the five other UC campuses are looking to develop tools that can provide an understanding of each campus landscape. This includes identifying existing innovative activities and seeing what is needed to improve their growth and strength. These ongoing efforts are aided by existing projects such as Emerging Technology Observatory's Open Source Software Research and Community Activity (ORCA), the NumFOCUS Map of Open Source Science (MOSS), and the Apereo Foundation's Open Source Software in Higher Education Report.[3] The data gathered by the UC-developed discovery tool will help campuses target outreach activities that promote open source sustainability and best practices while also curating a portfolio of open source projects that would be open to engagement with external sponsors.

Ultimately the UC OSPO network project team hopes to include all 10 UC campuses as well as affiliated national labs in these collaborative activities. An important part of this network's initial work will be to support local campuses building their own OSPO infrastructure. Even within the same system, there is no one-size-fits-all approach. While learning from the process undertaken by UCSC and other academic OSPOs, each campus will develop a unique structure that meets the needs of their individual campuses.

The successful creation of a UC OSPO network will likely hasten the proliferation of academic OSPOs. As part of one of the largest public research university systems, a University of California OSPO network has the potential not only to transform the UC system's approach to open source but also to set a precedent for other large university systems establishing their own OSPO networks.

To hear more about the open source activities at UC Santa Cruz and how the UC OSPO Network develops, check out the UCSC OSPO website at ospo.ucsc.edu and follow us at linkedin.com/company/ospoucsantacruz.

[3] See Open Source Science (OSSci), *The map of open source science (MOSS)*, OSSci; Patrick Masson, *Open source in higher education… a community report*, Apereo; and Zachary Arnold and Jennifer Melot (2023, July 11), *Data visualization: ETO open-source software research and community activity (ORCA)*, Center for Security and Emerging Technology.


Be bold. Be boundless.
Be a Baskin Engineer.
UC SANTA CRUZ
Baskin
Engineering
engineering.ucsc.edu

ChRIS *five years later*

The groundbreaking platform levels the playing field for advanced analytics and AI in medicine

An interview with ***Mo Duffy and Rudolph Pienaar***

conducted by ***Orran Krieger***

RESEARCH QUARTERLY

VOLUME 6:1

Red Hat

Interview

What if there were an open source web-based computing platform that not only accelerates the time it takes to share and analyze life-saving radiological data, but also allows for collaborative and novel research on this data, all hosted on a public cloud to democratize access? In 2018, Red Hat and Boston Children's Hospital announced a collaboration to answer this question, deploying the ChRIS Research Integration Service (formerly the Children's Research Integration System) on the Mass Open Cloud (MOC), making it possible to rapidly apply new and more complex computational tools to image analysis, reducing the burden on radiologists and potentially saving more lives. Leading this effort were Dr. Rudolph Pienaar, Staff Scientist at Boston Children's Hospital and Assistant Professor in Radiology at Harvard Medical School, and Máirín "Mo" Duffy, Senior Principal User Experience Engineer and Software Engineering Manager at Red Hat.

In 2023, the MOC evolved into the MOC Alliance, with production cloud services operated by Harvard and BU Research IT as the New England Research Cloud (NERC), and shortly thereafter ChRIS deployed on NERC. We asked Boston University Professor Orran Krieger, Co-Director of the BU Red Hat Collaboratory at the Hariri Institute for Computing and PI of the MOC Alliance, to interview Duffy and Dr. Pienaar about the developments with ChRIS. What follows is an animated conversation about the critical roles AI, cloud technology, and open source have to play in both clinical medicine and medical research.—Shaun Strohmer, Ed.

About the Interviewer

Orran Krieger is the Co-Director of the BU Red Hat Collaboratory at the Hariri Institute for Computing, PI of the MOC Alliance, and a Professor in the Department of Electrical and Computer Engineering at Boston University.

Orran Krieger: In 2016, AI pioneer Geoffrey Hinton said that he didn't know which jobs would be destroyed by AI, but radiologists were like the coyote who's already run over the cliff and hasn't looked down yet. He also claimed that, as a result, in five years there wouldn't be a job for radiologists.

So, was he right? Has AI solved all the problems? What is your position on the application of AI to radiology and is ChRIS all about AI?

Rudolph Pienaar: What is my position on AI? That's a very big question.

Orran Krieger: Specifically, is AI useful for radiologists? And what role do you see AI playing in the future?

Rudolph Pienaar: Of course it's useful. But before I answer, even that question needs to be taken apart. It's like saying, "Mathematics can solve these problems." You have to talk about the specific technique. Usually when people say AI these days, they're talking about neural network-based technologies. That's one tiny island in a lake in a continent on this huge planet of the field.

Neural networks ultimately are good at a limited set of problems. By luck and chance, those

WHAT IS CHRIS?

In its earliest form, ChRIS focused on making innovative image analysis available to clinicians who didn't have time to wrestle with a complicated interface. Dr. Ellen Grant, director of Fetal and Neonatal Neuroimaging Research at BCH and professor of Radiology and Pediatrics at Harvard Medical School, was frustrated by not having the latest research tools accessible for clinical practice. Dr. Grant brought in Dr. Pienaar to help create infrastructure allowing clinicians to take advantage of advanced image-processing technology from a desktop or mobile device, starting with a web-based front end for complex tools.

Dr. Grant also wanted a way for clinicians and researchers to tap into the potential of the huge amounts of data stored in a hospital database or PACS (Picture Archiving and Communications System). Radiologists spend thousands of hours reviewing hundreds of thousands of images to flag abnormalities in patient scans. ChRIS shifts that task to a model trained on image data, reducing a clinician's time to diagnosis by letting them spend less time shuffling through images and more time interpreting them.

Moving ChRIS from the high-performance computing systems at BCH to the MOC meant that clinicians and researchers could access complex analysis using multiple computers in the cloud. As a result, image analysis that could take a day using hospital compute resources would take only minutes on the open cloud.

While ChRIS's origins are in fetal and neonatal neuroimaging, the platform can be used for anything from genomics to electronic medical record text processing. For example, in 2020 Red Hat and BCH worked with DarwinAI to help develop COVID-Net, a suite of deep neural networks for COVID-19 detection and risk stratification via chest radiography. Find ChRIS documentation, plugins, code, and more at github.com/FNNDSC.

problems are pattern recognition, and pattern recognition is a huge factor in many problems we try to solve. So in terms of automating pattern recognition problems in radiology, AI is an extremely useful tool. It is a tool that is useful in a domain, and we have to understand what that domain is.

Orran Krieger: So AI is one of a set of techniques that can be very helpful as part of the whole workflow assisting the radiologist. And via ChRIS, you provide a tool that makes it possible to apply AI technologies where they're useful. ChRIS allows people to exploit AI, along with other techniques, to solve problems.

Mo Duffy: Like Rudolph said, it's like math. What problem are you solving: are you building a bridge or figuring out how much to pay for groceries? That's where I'm coming from as a UX engineer. If there is a user story it solves, great—bring it in. But the overall ChRIS platform is geared towards cloud compute, with a specific bent towards radiology, although it can be more generic if needed. That's the core piece: solving radiology workflows.

If we can plug in AI to make that better, that's great. But the problem to solve is that you have a patient who needs to be evaluated, the radiologist has a huge backlog, there are not enough radiologists on staff to do it in time, and you have to prioritize so the most needful patients get looked at first.

Orran Krieger: What is the problem radiologists and researchers developing computational tools for radiology have that ChRIS addresses?

Mo Duffy: I think it's that all of these open source tools are out there, but radiologists are not going to open up the Linux command line to use them.

Rudolph Pienaar: Yes. The biggest problem is that there's currently no easy way to inject computing into a clinical cycle. Because there's no bridge between the compute and the clinician, the connection never happens.

I've seen this many times: you have amazing tools on the computing side, but they don't magically jump that gulf to help inform a clinical workflow because the infrastructure hasn't been addressed or put in place. We've focused on building infrastructure to link computing in workflows end to end in the way radiologists use computing. The computing has to be

in the woodwork: the less a radiologist has to do with computing, the better.

Mo Duffy: Even just opening up a new browser window and logging into a new app—you've lost them.

Rudolph Pienaar: You've lost them. If you don't recognize that, you're never going to solve that problem. We've tried to build a system that will take a particular image, do some fancy computing on it, then put the results front and center in the radiologist's workflow, and they didn't have to do anything. Linking computing to clinical workflows is the first big way we're hoping to make a contribution.

Orran Krieger: What drove each of you to start working on this project?

Rudolph Pienaar: I'd seen so much good software created in my academic career that could make a difference in the real world just go to waste. You create stuff to get a paper published, and when the paper is done, you hope someone will read the paper, be inspired by what you did, and try to either recreate it or build something from it. But you built something yourself, and you threw it away. That was so wasteful to me that it became almost impossible to not think about solving that problem.

Once you start scratching at something, you realize how complicated it is. It is so hard to build an infrastructure. In our case, luckily, we started thinking about infrastructure at the same time technologies were coming online enabling containerization—cloud computing with OpenShift or OpenStack or Kubernetes. These were all bubbling together at the same time, and suddenly they could be connected.

The FreeSurfer plugin for ChRIS performs volume- and surface-based analysis of MRI data and facilitates 2D and 3D visualization of various segments in the gray matter of the brain.

Mo Duffy: For me, since I was in high school, I've used Linux. I went to an engineering summer camp and learned this creative development platform and I got very into it. During my undergrad, other people came to me to learn this platform. Then the company making the platform got bought out by another large company, and they shelved the

product. So I spent my college years learning a tool, and while the skills still transfer, all these assets I developed over my career were not opening. So I understood the benefit of open source.

Then my father had a stroke. As a user experience designer, I like to know, what is the problem to solve here? In this case, we have to figure out fast what type of stroke it is to know what kind of treatment is needed. How can we help radiologists prioritize patients in the most urgent need, or give radiologists hints if they have a backlog to get them through that backlog in the most efficient way possible? I was very motivated by this personal event, and I thought, "I get this, and I think open source can help."

Orran Krieger: There are lots of large proprietary companies in the medical industry. Why haven't they built those workflow tools into, for example, the radiologist's workbench or the MRI machines?

Rudolph Pienaar: I think a core reason is that medicine is more about the human making the decision based on the data in front of them, as opposed to some other person or process taking the data and adding value to it. It's not part of the DNA of a doctor to rely on an external thing to make a decision.

Mo Duffy: When you employ an external thing—say it's AI—but it's proprietary, it's hard to trust it because you don't know what it's doing. That's why open source is important in this space. If we're going to build tools radiologists can trust, it cannot be a closed black box. With open source, you have some control. You can understand what is going on under the hood. If you need to trust something, you can query it.

Rudolph Pienaar: Even if you don't do it yourself—I don't think a doctor's ever going to look at some code—at least it is out there and can be examined, and if there is something wrong it can be flagged.

Mo Duffy: Also, a vendor-hospital relationship is a one-way thing. But when the tooling is open source, you can collaborate. You can work together, share what you've learned, and put it back into the tool to benefit everyone.

That's why open source is important in this space. If we're going to build tools radiologists can trust, it cannot be a closed black box.

Orran Krieger: How do you see ChRIS enabling collaboration—across hospitals, between hospitals and researchers who want to develop techniques for analyzing images, in open source communities?

Rudolph Pienaar: That's a good question. We have tried very hard to make the system turnkey simple. It is super easy to get started with ChRIS today for anyone, no matter where they are. You don't have to be extremely technical; you just need to be able to read some instructions and you are up and running. That's a huge win.

We also try to lower that barrier of entry for making apps for ChRIS. It's much simpler to write a ChRIS app than it is to write an iPhone or Android app. You can easily run a ChRIS instance on any infrastructure, and once it's running, it's very easy to send and process data. As a user, you have visibility into that in a very straightforward way, and then you can compute. You can write. You can develop a program on your local laptop. And the same program you've run on your laptop with your local ChRIS will run on ChRIS in the cloud, so you can process a huge set of data without having to download it.

I've jumped over a lot of ground in those statements: there's security, there's safety. Assuming those are being addressed, the infrastructure makes it very easy to do these things.

Mo Duffy: It meets the researchers where they are, rather than bootstrapping them into becoming production software developers. We can say, this is how you write the code. We'll work with that, we'll help you containerize it and make it cloud deployable, which is not something they would have the ability to do otherwise.

Orran Krieger: So one user group is physicians, and they need a pretty user interface or something invisible.

Then there are developers who want to write things as a scientific program. Then to make this thing deployable, there's this Red Hat technology of containerization, cloud services, and

other stuff to make those things meet in the middle. Is that accurate?

Rudolph Pienaar: Yes, very accurate. ChRIS runs very well on a lot of cloud tech, especially OpenShift. What we're exploring at the moment is the ChRIS app. There's no restriction on what that app has to have inside of it. A researcher like me might write a ChRIS app with some kind of AI algorithm. You can develop quickly, and you can then run it on the cloud.

And, nothing prevents you from scaling much higher. You could take that AI app and retool it, as Mo mentioned, in a more scalable cloud infrastructure—let's say, OpenShift—so it becomes part of the OpenShift infrastructure. Running the app now becomes very simple. Instead of having all the AI inside of it, it just has a few function calls out to the OpenShift infrastructure.

What's a win-win is that from the moment the scientific researcher writes their AI app, they can start running it and get results while they spend three weeks or two months rewriting it at scale for OpenShift or OpenShiftAI. There's no dead time: you can still be developing your code and testing it, it can be delivering results, and you have a path to run it at a massive scale.

Orran Krieger: You're at a very rich hospital. What does ChRIS offer to poor hospitals, for example in places in Africa, where you're originally from, or hospitals in the rural United States?

Rudolph Pienaar: This is where open source tech and infrastructure make a level playing field. A hospital in a poor area might not be able to pay millions of dollars for the bespoke kind of radiology solution Boston Children's Hospital has, but it certainly is possible to buy a bunch of PCs and deploy them on premises. Then you can throw OpenShift on them and install ChRIS on that, and you have essentially the same computing infrastructure that a place like Boston Children's Hospital can have. Or, maybe easier, you can offload all this to a pure public cloud like the MOC and benefit from hosting at scale. And since the MOC offers huge benefits for nonprofits and also runs the same OpenShift as you would on prem, the deployment complexity and experience is the same no matter where you choose.

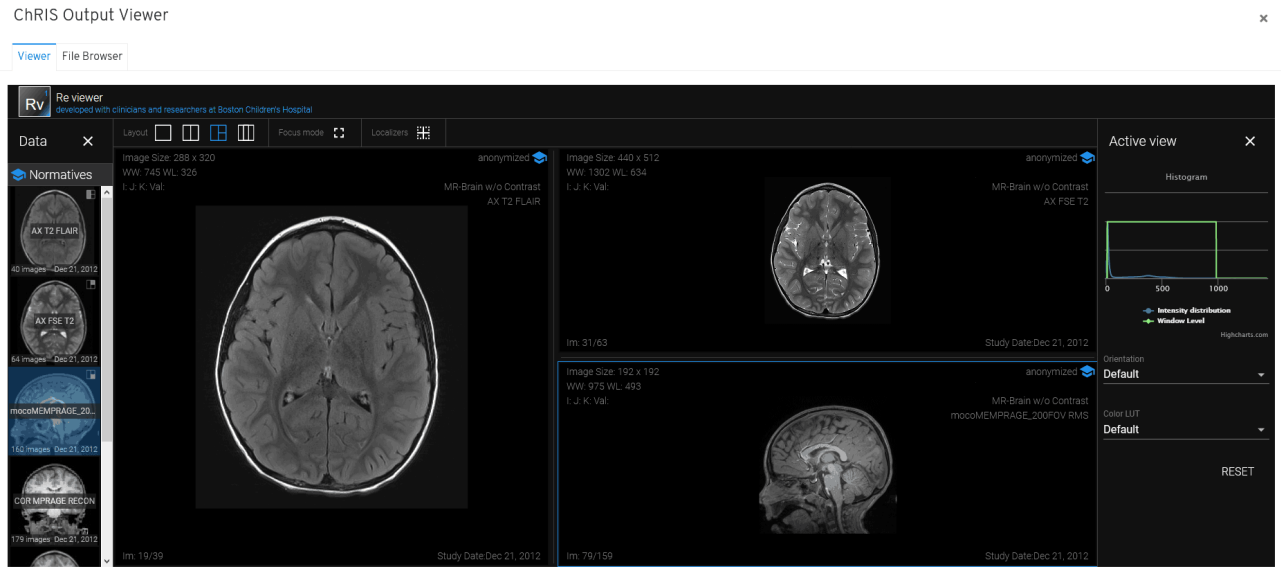
You could even start for free, running either OKD or the free developer version of OpenShift, then potentially license that further up and get more tools or solutions. You have that scalability path going upwards, but you don't have to have a huge outlay at the start.

Orran Krieger: Does ChRIS offer value for collaboration across hospitals? Each hospital has its own infrastructure, which creates silos.

Rudolph Pienaar: Yes, silos across hospitals, or even within a hospital. They're just an artifact of how hospitals are run and put together. If I'm running analysis on a piece of data, there's no discoverable way for anyone else to know I've done that. Everyone is sitting on their own laptop, and they aren't connected at a fundamental level, at the data level.

If there were a hospital-wide ChRIS, then in theory all the data is seen by this one platform. It would be possible to say, "Oh, Orran did this analysis

If multiple hospitals start to host data on ChRIS on the MOC, we enable the ability to compute at scales hitherto unheard of.



ChRIS pulls brain MRI images from the hospital PACS into the main workflow for analysis.

three weeks ago, which is exactly the same analysis I want to do today." It's going to give the same results that took 10 hours of computing time. There's no need to waste those cycles again. Having data grouped together in a unified platform like ChRIS makes that problem extremely solvable.

This of course leads ultimately to the MOC, where, if a hospital hosts their radiology IT on the MOC then, using ChRIS, the sharing problem becomes moot. Not only within one hospital, but if multiple hospitals start to host data on ChRIS on the MOC, we enable the ability to compute at scales hitherto unheard of. If the MOC becomes HIPAA compliant, that will really unlock its potential.

Mo Duffy: Just being able to access the PACS (Picture Archiving and Communications System) in a visual way and see, "Oh look, some other people pulled this file. I don't need to re-pull it to my local PC because somebody already pulled it to ChRIS"—that's a simple problem but solving it makes a big difference.

For the areas with limited technology, the needs are very simple. Just having something like the open source PACS server is huge. It's game-changing for some facilities.

Rudolph Pienaar: That's an extremely good point. Just being able to get to your data in a simple way already changes everything, and we can do that today with ChRIS and with open source solutions running on OpenShift. That opens up whole avenues of potential clinical work.

A hospital might have, say, an MRI scanner, but they can't afford the licensing or the fees to save its data to a PACS, or they aren't able to buy a PACS. Instead, someone scans an image and burns it to a flash drive, takes it out of the scanner, walks across campus to their office, plugs it into their PC, then runs whatever janky software might come with a scanner to look at the images. No organization, no databasing—just a bunch of flash drives floating around, and no one knows what was scanned yesterday because it's too much to keep track of—all for the lack of a free, open source solution they could deploy if it were easy and possible to do so.

We're at the point now where it is easy and possible.

Orran Krieger: What's the relationship between ChRIS and an open source PACS solution?

Rudolph Pienaar: We combine it into the deployment model of ChRIS. These days you can deploy ChRIS with just a few files and a few lines of YAML in a script relying on Helm charts. If you have OpenShift and you have a Helm chart, which we provide, it installs ChRIS, and you get all of the stuff around it in that same deployment. And as part of deploying ChRIS you also get a connected PACS with which ChRIS can immediately communicate.

Mo Duffy: We tried a ChRIS store: an online catalog of ChRIS plugins where people could list plugins themselves. Then we built it into ChRIS as part of the core platform, so when you go onto the ChRIS server you can see the plugins available, or you can look at another organization's ChRIS server to see what feeds, workflows, or pipelines they use. The output has been made public, so you can see, "Oh, there's this this plugin, and they ran this this feed with it, and this was the output." If this looks like the kind of analysis I'd like to do, I will install that plugin.

Now, say you're a radiology researcher presenting at a conference and you have some new technique to show off. You want people in the audience to say, "Hey, I'd like to vet that technique and try it on my own data and see if it still pans out." What if, in your presentation, you had a QR code that goes to your ChRIS server? You've made this feed public on your ChRIS server to support your research, and any researcher watching your talk or reading your paper could scan it, visit your ChRIS server, download the same plugin on their own ChRIS server, then attempt to reproduce your research with their own dataset. All the tech stuff that's challenging for a typical researcher is taken care of by the platform.

I don't think we're there yet, but it's on the roadmap as one way we could enable cross-institution collaboration

Rudolph Pienaar: Definitely. We also have our flagship reference public ChRIS running on the MOC-Alliance production cloud (NERC) right now. You can go to that instance and see all of its plugins. It functions as a de facto worldwide ChRIS store.

But to underscore Mo's point, to make collaboration easier, you need to reduce the work any one person has to do. If you write a ChRIS plugin and I want to run it, the hardest thing I need to do if I have a ChRIS too is type in the URL of your plugin. I am guaranteed it will run on my ChRIS, it will be the exact version you wrote, and if I feed it the same data you published on your site, I will see the exact same results.

Orran Krieger: That's very powerful. What are some things integrated into the workflow today that clinicians are using ChRIS for?

Rudolph Pienaar: We have a couple of projects that are already delivering value. One is a project that looks at leg X-rays and measures the lengths of the legs. That doesn't sound extremely sexy, but it makes a very big difference, because the tools vendors provide don't do any of that automation.

A clinician looking at the X-ray of a kid's legs is trying to figure out if they are the same length, because if they aren't, they need to figure out what's going on. With the tools currently available, a clinician has to pull up the image on screen and use on screen tools to manually measure. So if I were doing this, I click on the measurement tools and with my mouse move to the top of the hip joint, eyeball it, click, pull the line down to the knee, eyeball it, click, and it will give me a measurement of what that distance is. I rinse and repeat for both legs, upper leg, lower leg, and I end up with all these numbers on my screen. Then I either look at it in my head and try to figure it out—not a good idea—or I use my phone and add them together, or write on a piece of paper, or whatever. It takes easily 4 or more minutes.

Mo Duffy: It's very manual, and it's a very fine-motor task. It takes a lot of effort.

Rudolph Pienaar: It's a problem that you would think, "hey, this could be done by a computer," and guess what? It can. A clinician at BCH wrote a prototypical solution (pl-lld_inference) using off-the-shelf AI-python libraries that gives very good results. We cleaned that up and packaged that as a ChRIS plugin. Still, this one plugin just does one thing: tries to find the ends of leg joints.

To add meaning, you need an ensemble cast. Different actors each do their small part, but taken together they solve a problem. We built all these and can describe what they need to do in a simple YAML workflow that ChRIS can interpret, deploy, and execute. So one plugin queries the hospital PACS and downloads the image. Another converts the hospital imaging format to JPEG because the AI algorithm needs JPEG. The AI plugin loads the JPEG and outputs another JPEG

showing where it thinks the landmarks are. Another program picks up that output, looks at the landmarks, then measures the distances between them. Then another program picks it up and draws where those distances are on the image and writes a little summary table of them on the side, and yet another program takes that JPEG and reconverts it back to a medical image based on the original medical image. Another program picks it up and pushes it back into the hospital database.

Each one of those little programs is a ChRIS plugin available in the ChRIS catalog. Technically each is a standalone Linux container, and ChRIS can string them all together. In fact, ChRIS can even run each one on a different cloud, or on prem, or any combination, all as part of the same workflow. From the doctor's point of view, they just go to their workstation, call up the patient data and then, as if by magic, this ChRIS-processed image is also right there waiting for them amongst all the others for that patient. All they have to do is look at that annotated image and within a second or two, they will know whether it found the right positions. Within 10 seconds, they can decide whether they need to think about a potential medical issue, versus spending five minutes painstakingly doing these manual drawings. That's a real-world example we're solving right now.

Orran Krieger: What role does the Mass Open Cloud and its production environment play in this?

Rudolph Pienaar: They play a fundamental role, especially when it comes to breaking down those barriers and those silos and bringing collaboration to the next level.

Even if you imagine a world where all these different hospitals have their own ChRIS, are people going to send data easily from one ChRIS to the next? Are they going to trust it? You need a public-facing cloud provider offering a compelling reason to host data and compute there. That is what will bring these pieces together. With more data and more compute, the on-prem IT becomes even more complex, which is why at a certain level having a provider like the MOC becomes not only natural, but more importantly allows for scaling at levels beyond any single hospital. That opens the door for innovation at that next level.

Mo Duffy: When you're talking about proprietary versus open source solutions and building trust, it's the same with a cloud platform. If it's a closed proprietary cloud platform, versus an open, publicly owned platform, there's a trust difference, and that matters.

Orran Krieger: So if we can create an open public cloud, we reduce the barriers for hospitals to use this technology.

Rudolph Pienaar: Exactly. And you still have ownership of data you put in a public cloud, but it makes it very possible and easy for you to share it.

Orran Krieger: But hospitals do share data when they do collaborations, right?

Rudolph Pienaar: It is extremely cumbersome and idiosyncratic. In fact, it is so bureaucratic and so complex that it's oftentimes easier to not do it.

Mo Duffy: That's where something like ChRIS makes a difference. If each institution has its own ChRIS, and you share the pipelines, the feeds, and how you built it, then each institution can run it on its own data in-house and not have to worry.

Rudolph Pienaar: Exactly. That's certainly a very attractive feature of a platform like ChRIS.

Here's a scenario: I want to get medical data from somewhere to develop a new technique, then share the results. Or there's a GitHub repo of some interesting technique I want to try on this dataset. Without a platform like ChRIS, it's very hard. You have to find the data. Then you have to do some idiosyncratic database magic to get the data, you have to download it to your local computer. You have to go to the Github repo of the thing you want to run and figure out how to run it. Maybe it'll work. Maybe it won't. How much work will it be to get it to run? You don't know any of these things beforehand. Then you have to construct your own environment in your machine or figure out how to run it on your local cluster.

With ChRIS, that infrastructural cost dials down significantly. ChRIS already speaks to the arcane image databases you find in hospitals and provides you, the user, with a simple intuitive way to search and collect from this. If the program you want to run is a CHRIS plugin, downloading from the database to your ChRIS and running some cool analysis is literally one URL away, and

you know it will work. Sharing the results of that—again, drop dead simple.

In my experience in the field of scientific computing, things make or break on the infrastructural costs. That's not just money, but just the amount of work you have to do. If you dial that down, suddenly innovation becomes way easier.

Orran Krieger: What's exciting to me is that scientists developing new image processing techniques can make a discovery, and this discovery could be available to all the hospitals of the world the next day. With ChRIS on an open public cloud, if there's a group at a top hospital that develops a technique, they can disseminate and publish it in an executable way to affect real patients in real time—that's pretty cool. I didn't think about it that way until this conversation.

There's been a large community of people that have played a role in ChRIS. Anyone you want to give credit to?

Rudolph Pienaar: I want to give a huge shout out to my BCH team: Jorge Bernal, Gideon Pinto, Sandip Samal, Jennings Zhang, and Chan-Heng Hsiao. Of course to Ellen Grant, my long-time collaborator without whom ChRIS would have never existed. There are so many other folks, too many to name—eight semesters' worth of BU and NEU CS graduate students who have contributed, three years' worth of Outreachy Interns who helped on the project, lots of folks at Red Hat, especially Mo (of course), Hugh Brock, so many other talented folks who have dropped in with their valuable time. It has been and is quite amazing.

TAKING CHRIS TO THE EDGE

Through a partnership with the College of Charleston in South Carolina, computer science students are working with Red Hat engineer Isaiah Stapleton to enable ChRIS on edge computing devices. Meet ChRIS in a Box.

Bridging the gap between ultramodern cloud-based processing capabilities and on-premises edge computing resources has significant potential to foster innovation and improve patient outcomes. ChRIS in a Box (ChBox) meets that need by allowing you to deploy ChRIS software to any edge device, even Raspberry Pis. ChBox, a joint project of BCH and Red Hat led by Senior Solution Architect Raghuram Banda on the Red Hat side, is a self-provisioning system that can schedule containerized compute on the edge, autonomously analyze patient data, and then push results back into clinical systems operating at the edge. College of Charleston students and I are collaborating on an edge-device-provisioning web application tailored for ChBox.

ChBox operates by deploying components of the ChRIS application as containers, leveraging tools such as Podman and Microshift. By simplifying the provisioning process and using containerization technology, ChBox streamlines edge device setup and facilitates medical analytics tasks, particularly in regions where resources are limited. With ChBox, a clinician working in Rwanda, for example, where there might be a shortage of medical resources, can use the same tools available to the world's leading hospitals to help clinicians diagnose patients.

The user-friendly edge-device-provisioning website developed with the students will enable users to effortlessly provision edge devices and install ChRIS software onto them. It simplifies provisioning with an easy-to-use interface that allows users to select which edge device they want to provision software for, generating a link to an ISO image that users can put onto a USB drive and plug into their edge device to install the ChRIS software. The envisioned application will serve as a gateway for hospitals and healthcare facilities to integrate ChRIS capabilities seamlessly into their infrastructure.

Students Julia Kempton, Siah Thomas, Channing Smith, and Zi Yi Xiao are contributing to this project. Through their work they are building skills in full-stack web application development, Git/GitHub, agile practices, and more, positioning them to succeed in industry after graduation.

—Isaiah Stapleton

Says who?

/Keep your options open
redhat.com/options

Red Hat

A AWS

B Azure

C Google Cloud

D All of the above

/Keep your options open
redhat.com/options

Red Hat

MOC ALLIANCE

MAKING THE CLOUD LESS, WELL, CLOUDY

The Mass Open Cloud Alliance (MOC Alliance) is a collaboration of industry, the open-source community, and research IT staff and system researchers from academic institutions across the Northeast that is creating a production cloud for researchers. Of course, a collaboration is only as good as its collaborators.

Follow the MOC Alliance as they create the world's first open cloud.

@mass-open-cloud

www.massopen.cloud

contact@massopen.cloud

Housed at:

Boston University Rafik B. Hariri Institute for Computing and Computational Science & Engineering

BOSTON UNIVERSITY

RESEARCH QUARTERLY

VOLUME 6:1

Red Hat

Feature

Moving ecological forecasting from supercomputer to cloud: why and how

New event-driven architecture enabled researchers to move the PEcAn platform to the New England Research Cloud and increase scalability.

by Christopher Tate

Near-term ecological forecasting can help communities make better decisions and prepare for extreme weather events and changes in the environment. Use cases include forecasts of infectious disease outbreaks, increases or declines in animal populations, or the impact of environmental events on agriculture, forestry, or other industries.

A large and active science community has formed around ecological forecasting, and scientists in several different countries are interested in experimenting with the ability to make forecasts based on large sources of local environmental data. To make these capabilities available to as many public researchers as possible, the forecasting tools must provide open, accessible, reusable, and scalable community cyberinfrastructure that can make large numbers of ecological forecasts on a repeatable, frequent basis. A project underway at the Red Hat Collaboratory at Boston University is addressing that challenge by developing a cloud-native workflow that provides asynchronous, event-driven, and distributed computing and resource management for these large-scale science projects.

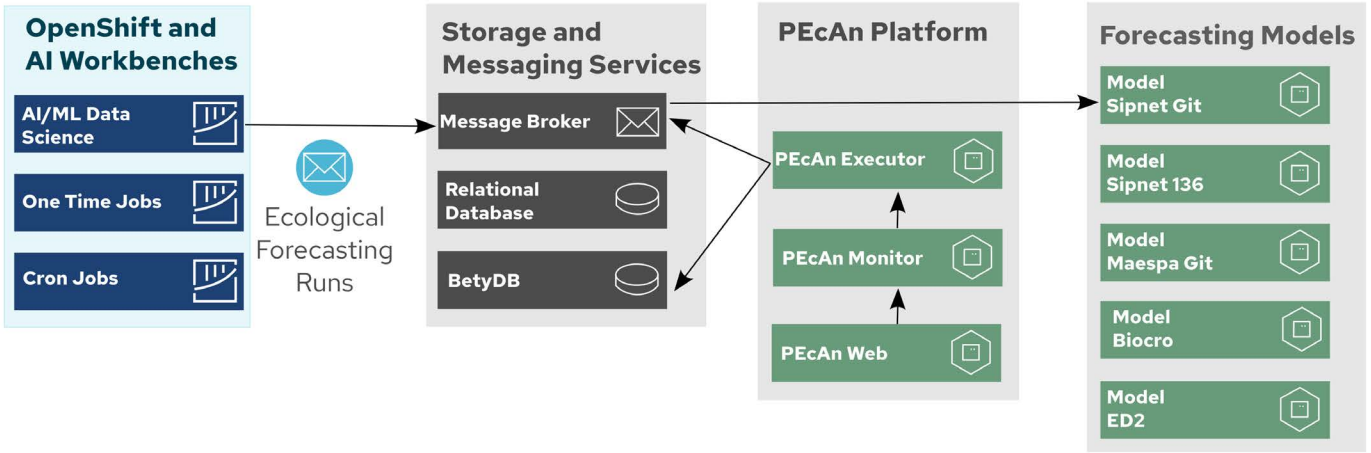
PROJECT BACKGROUND

As a software developer at Red Hat Research, I work with students and professors to advance their research goals with open source technology. In this project, I'm working with the ecological forecasting team at Boston University, including Professor Michael Dietze, PhD students including Dongchen Zhang, and a team from the BU Software and Application Innovation (SAIL) lab, including Associate Director of Programs and Product Management Jeff Simeon, Associate Director of Engineering Greg Frasco, and Software Engineer Shashank Karthikeyan. The team encountered challenges collaborating on an open source software platform running on a shared high-performance supercomputer at Boston University. Switching to a more event-driven cloud architecture enables autoscaling of forecasting jobs on multiple nodes concurrently.

We began the project in January 2023 with a Research Incubation Award from the Red Hat

About the Author
Christopher Tate is a principal software engineer for logging, metrics, alerts, AI/ML, and data-driven research projects in the New England Research Cloud (NERC) environment.

See *"Prototyping a distributed, asynchronous workflow for iterative near-term ecological forecasting"* on the Red Hat Research website.



PEcAn platform on NERC OpenShift

Collaboratory. We reviewed potential event-driven services available to us in a containerized cloud environment and developed an architecture that would work to scale continually updated predictions about the future state of ecosystems over days or years. Scaling near-term ecological forecasting in this way enables the development of services that allow communities to anticipate environmental challenges and improve decisions on actionable timescales. We believe it will also allow researchers to accelerate scientific discovery and answer some fundamental research questions about the predictability of nature.

We developed an architecture that works inside an open source cloud environment like Red Hat OpenShift. For data analysis, we used an established open source project called the Predictive Ecosystem Analyzer (PEcAn). The PEcAN community has already developed containerized data science models, database enhancements, and tools to display, monitor, and execute ecological forecasting models. What we needed on top of that was a cloud environment, a message broker supporting the Advanced Messaging Queuing Protocol (AMQP) to receive messages to run forecasts, and a scalable PostgreSQL database with built-in geolocation features to store ecological forecasting data.

Our ultimate goal was to deploy our solution on the New England Research Cloud (NERC), but until the production cluster was ready for use, I worked on the solution on my own computer using OpenShift Local. By trying out the existing PEcAn Helm charts, we developed a reusable way to deploy the PEcAn project easily into OpenShift. Together with the SAIL team, we developed additional open source infrastructure-as-code for the project that was reusable for developing and deploying our project on our own computers, as well as in future cloud environments.

One early challenge was our discovery that parts of the PEcAn Helm charts, which in the past have been deployed only on Kubernetes, were violating security constraints built into a Red Hat OpenShift Container Platform by default. I contributed some updates to multiple PEcAn Helm charts so that service accounts could correctly deploy the PEcAn containers. The PEcAn open source community accepted those updates to their repositories, and we then had a working cloud solution on OpenShift Local.

ONBOARDING TO NERC

Three months into our project, NERC opened up its new OpenShift Container Environment for research projects. We were quickly able to deploy all the same components working in OpenShift Local to NERC and show the platform running forecasts in the production-ready cloud environment, which was a very exciting moment.

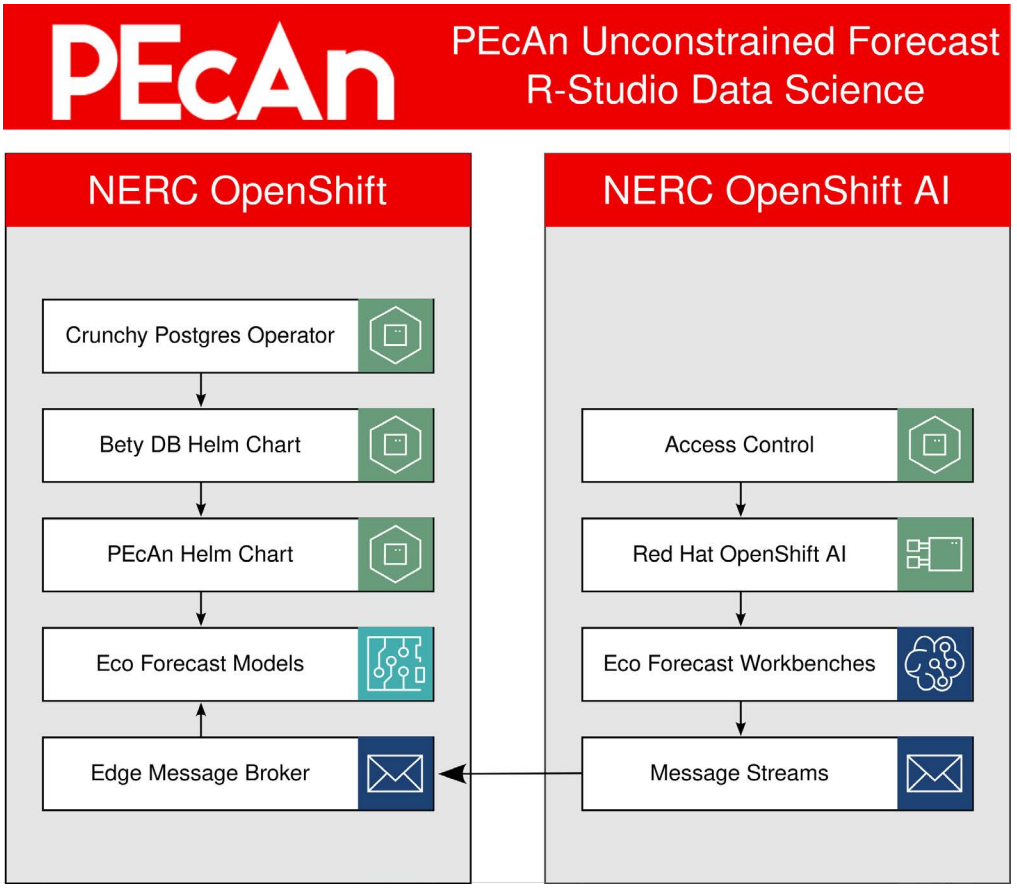
Students at BU run ecological forecasting code written in the R language, so to replicate that environment we next needed to enable RStudio in NERC, which was made

possible by OpenShift AI. Our team carefully prepared two pull requests that would enable ecological forecasting in RStudio in OpenShift AI on the NERC: 1) a new PEcAn Unconstrained Forecast container image based on an RStudio Jupyter Notebook container image that loads additional R dependencies and compiles the PEcAn source code and 2) OpenShift image streams for RStudio, as well as the PEcAn Unconstrained Forecast, that had the the right namespace, labels, and annotations needed for OpenShift AI Workbenches running on NERC.

The NERC Team released OpenShift AI into the production OpenShift Cluster in August 2023 and later merged our project's image streams into the computing environment in October 2023. From that point on, the science team was remarkably more comfortable working in the NERC environment. Our team no longer needed the skills of an OpenShift Admin to work in R Studio and run ecological forecasting. Being able to run the same workbench at the same time enabled all three teams—Red Hat, BU, and SAIL—to work together on the project. I was amazed at the productivity boost of our team and results from that point on. Having a friendly user interface in the cloud made a big difference.

EVENT-DRIVEN ECOLOGICAL FORECASTING AND SCALING

With OpenShift AI available in NERC and the new R Studio image we built, we were able to develop the new event-driven workflow in the PEcAn code and test the workflow all in the cloud. We requested that the Red Hat Custom Metrics Autoscaler OpenShift Operator be deployed to NERC to allow forecasting model pods to scale, according to the number of AMQP messages sent at one time. This works very well for running multiple models at the same time, and it's event driven. However, PEcAn model pods, which were originally developed in a different HPC enviroment, required a shared filesystem. Red Hat Engineers and Boston University students worked together on a solution to send necessary files for each job to the right container in a cloud-based way.



OpenShift and OpenShift AI components in the NERC PEcAn implementation

Professor Dietze introduced us to his new branch for HF Landscape Unconstrained Forecasts, which has not been merged into the main branch yet. I created a branch off Dietze's branch that replaced hard-coded paths on the high-performance supercomputer with cloud-friendly environment variables. Dongchen, who is very experienced in high-performance R applications and PEcAn, has been working on more improvements. In Dongchen's branch, he has been smoothing out the bug fixes and improvements for integrating ERA5 (European centre for medium-range weather forecasts Reanalysis, 5th generation) environmental reanalysis data and observation

MEET THE ECOLOGICAL FORECASTING TEAM

Jeff Simeon
BU SAIL

Greg Frasco
BU SAIL

Shashank Karthikeyan
BU SAIL

Dongchen Zhang
BU PHD Student

Michael Dietze
BU Professor

Christopher Tate
Red Hat

prep functions, as well as adding features for message-driven RabbitMQ job forecasts. It's worth noting that this kind of long-term collaboration would not be possible without open source software.

There are a lot of important technical changes built into our branches. We built the HF Landscape branch of PEcAn into an R-Studio Jupyter Notebook Image called the PEcAn Unconstrained Forecast image and deployed the PEcAn Unconstrained Forecast image to Red Hat OpenShift AI in NERC. We ran the newly updated download R script for Harvard Forest Meteorological data to download over one year's worth of data to our workbench persistent volumes. We updated the SDA Workflow for North America R scripts to process the HARV MET data, and Dongchen updated the SDA Runner script to run in the cloud.

Finally, we had to work through the challenge of PEcAn being monolithic software meant to run on one giant computer with tons of storage. Since data science projects involve large nested directories of file data, we developed an rsync strategy between containers running in OpenShift to rsync files to a model pod as part of our event-driven strategy. From an OpenShift AI workbench, we can send a message to a forecasting model pod. This also triggers an rsync operation that copies all the relevant files from the workbench pod to the model pod. The model pod receives the files and the message and runs the ecological forecasting model on the data. The model pod then sends the files and additional data back to the workbench pod that triggered the message. The pod rsync operation is surprisingly fast and effective for this transfer. This may not be the long-term solution for event-driven ecological forecasting, but it has worked very well for us given our time constraints and limitations to upstream adoption of our research this year.

FUTURE MILESTONES

The next step for our team will be developing and deploying an asynchronous, event-driven scheduler that will elastically launch data ingest containers. This will enable scaling our prototype to multiple sites, more data constraints, and a collection of models. Future applications could extend this system to additional forecast workflows, such as water resources, biodiversity, zoonotic disease, or invasive species.

This project has been the fruit of successful collaboration among ecological forecasting experts at BU, faculty who provide high-level support, and students with technical domain expertise, in addition to industry know-how from software developers like myself. This combination, plus the computing resources made available to researchers through NERC, allowed us to tackle the engineering challenges of running the PEcAn platform in the cloud.

ⓘ To learn more about the project, join us at github.com/PecanProject or pecanproject.slack.com, or contact Professor Dietze.

Red Hat

RESEARCH QUARTERLY

VOLUME 6:1

Feature

About the Author

Luis Tomás Bolivar

is a Principal Software Engineer at Red Hat, Spain. He is currently working in the Ecosystem Engineering group and on research activities with a focus on cloud computing in general, and automation, networking, and AI in particular. He holds a PhD in Computer Science (University of Castilla-La Mancha, Spain) and has been involved in several EU projects.

About the Author

José Castillo Lema

is a Senior Software Engineer at Red Hat working with the Telco 5G performance/scale team. During his MsC and PhD studies, he worked on QoS routing in SDN and NFV Management and Orchestration. Has been teaching postgraduate courses for the last 6 years.

Unleashing the potential of Function as a Service in the cloud continuum

The PHYSICS project demonstrates the value of the FaaS paradigm for application development and data analysis. Here's how we enhanced the infrastructure layer.

by Luis Tomás Bolivar and José Castillo Lema

The difficulty of scaling, optimizing, and maintaining infrastructure makes cloud computing too complex or resource-intensive for many developers and data scientists. The Function-as-a-Service (FaaS) model (often called serverless computing, generically) allows users to run certain types of applications in a modern, scalable, and cost-effective way without the added complexity of maintaining their own infrastructure. The PHYSICS project (oPtimized Hybrid Space-Time ServIce Continuum in faaS) aims to unlock the potential of the FaaS paradigm for cloud service providers and application developers in a cloud-agnostic way.

PHYSICS brought together a consortium of 14 international partners leveraging a €5 million Horizon Europe grant from the European Commission, including use case leaders in e-health, smart agriculture, and smart manufacturing. Engineers from Red Hat took a lead role in the infrastructure layer, adapting and enhancing tools in the Kubernetes ecosystem for scaling, energy awareness, and multicluster automation, including automatic cluster onboarding and configuration of PHYSICS components on top.

This article provides a brief overview of the PHYSICS project and initial milestones before detailing our most recent work on the infrastructure layer of the project.

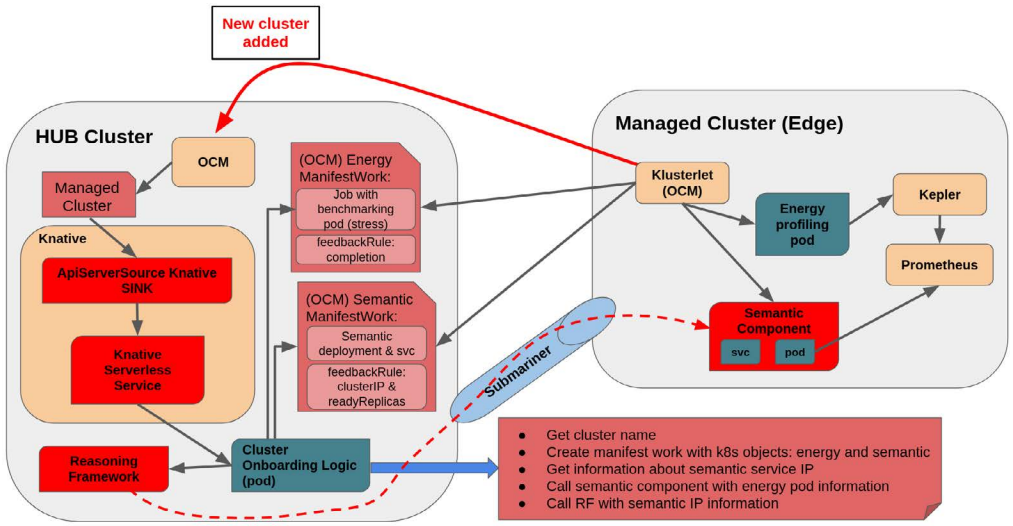
PHYSICS 101

PHYSICS facilitates the design, implementation, and deployment of

advanced FaaS applications, using new functional flow programming tools that harness established design patterns and existing cloud/FaaS component libraries. One of the key outcomes of the project is a novel Global Continuum Layer (distinct from the infrastructure layer) to facilitate efficient function deployment across diverse clusters.

The Global Continuum Layer is a set of PHYSICS components that optimize key application objectives at the same time, such as performance, latency, and cost. Use cases developed with industry partners include a smart manufacturing system to optimize production pipelines, healthcare software using machine learning (ML) models to monitor the health of individuals and analyze anonymized collected data, and a solution for near real-time greenhouse management that is responsive to dynamic conditions.

In our midpoint progress report, we described how visual flow programming and ready-made patterns can enhance function development and explored how ready-made patterns can enhance abstract function development. We also presented the Load Generator Metrics tool, which was developed to evaluate the performance of the different functions so that other PHYSICS components (such as the global orchestrator or colocation engine) could make optimized decisions about function placement across and inside clusters. (For a more detailed view, see the research day recording.



Overview of cluster onboarding components and interactions

ADVANCED PHYSICS

In the second half of the project, we identified opportunities for extensions and additions in the infrastructure layer. Our main working areas were threefold:

- Multicluster automation, using the Open Cluster Management API and Knative
- Energy awareness (with RYAX), using Kubernetes-based Efficient Power Level Exporter (Kepler)
- Scaling, using Kubernetes Event-driven Autoscaling (KEDA)

Multicluster support

Onboarding new clusters is an important challenge addressed by PHYSICS. Even given Open Cluster Management (OCM), extra PHYSICS components need to be installed and configured when adding a new cluster. Beyond deploying the new services, you need to:

- Connect them to the relevant PHYSICS components in the hub
- Create a benchmarking load in the added cluster to gather base performance and energy consumption data

When onboarding clusters into OCM, central and remote cluster configuration is crucial. The first step before deploying functions on the cluster is properly configuring them and obtaining enough information (semantics) about them to make wise decisions concerning placement. This involves connecting PHYSICS components in the added clusters with other PHYSICS components and applications at the hub.

The Open Cluster Management API and components serve as

the foundation, adding various Kubernetes objects in the OCM ManifestWork such as pods, services, and ServiceExports (for Submariner support). This also includes specific Custom Resource Definitions (CRDs), for example, the Workflow CRD introduced by PHYSICS to abstract the information related to the functions so that they can be deployed on the relevant FaaS engine—in our case, either OpenWhisk or Knative.

PHYSICS uses Knative capabilities for event-driven and serverless operations, which optimizes resource usage. When a remote cluster is added to the hub via OCM, a ManagedCluster object is created. The Knative APIServeSource then invokes the Knative Serverless Service upon receiving the event. The cluster onboarding pod, running as a Knative Serverless Service, processes the request, configures the edge cluster, and generates a benchmarking load. Specifically, the cluster onboarding pod:

- Obtains the cluster name.
- Creates an OCM ManifestWork, which includes the definition of the semantic deployment and its associated service. The Klusterlet agent in the remote cluster is in charge of creating the local resources in that cluster.
- Waits until the deployment is ready and obtains its service IP by using the OCM feedbackRule.
- Creates another OCM ManifestWork, which includes a Kubernetes Job that will generate some benchmarking load in the managed cluster. Again, the Klusterlet agent creates the Kubernetes Job in the remote cluster.
- Waits until the job is completed using the OCM feedbackRule.
- Calls the semantic service endpoint, leveraging Submariner to reach the semantic service IP. This provides information about the previous job executed to gather energy consumption and performance metrics to score the cluster.
- Calls the reasoning framework to provide the information about the semantic service IP, so it can start requesting semantic information from the new cluster.

Extra configurations can be added easily as part of the cluster onboarding logic component or even created as extra Knative Serverless Services that react to the same events and perform other actions in parallel.

> Kepler integration allowed us to estimate energy scores for new clusters, enhancing our understanding of energy consumption.

Energy awareness

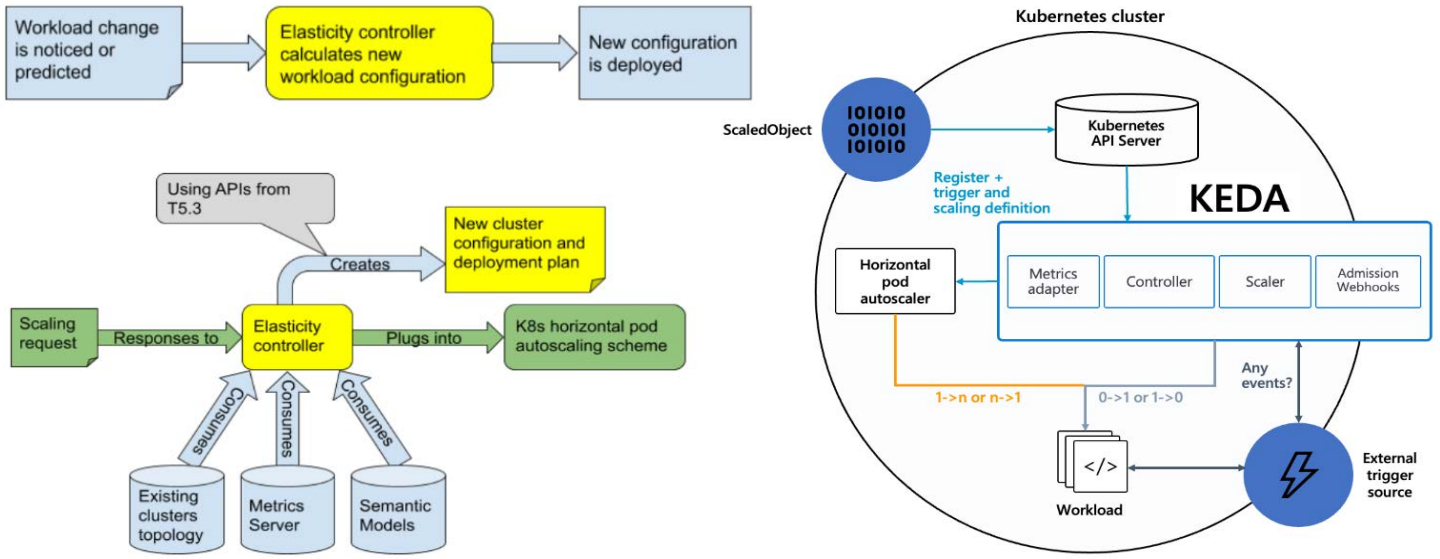
Kepler offers accurate energy estimates and detailed reporting of power consumption. It harnesses an extended Berkeley Packet Filter (eBPF) approach to attribute power consumption to specific processes, containers, and Kubernetes pods, running custom code in the Linux kernel (or other operating system kernels) to obtain the metrics to fuel ML models that estimate energy consumption.

PHYSICS selected the Kepler project to acquire energy-related information crucial for its components, including the scheduler. We integrated Kepler by incorporating its metrics (via Prometheus, an open source monitoring toolkit and one of the earliest Cloud Native Computing Foundation projects) into cluster onboarding and the PHYSICS semantic component. This integration allowed us to estimate energy scores for new clusters, enhancing our understanding of energy consumption.

Within the scope of the PHYSICS project, we actively engaged in Kepler's upstream development. Our collaboration involved:

- Identifying and resolving critical issues hindering nested environment utilization, i.e., when operating on top of virtual machines in public cloud providers (AWS or Azure)
- Making Kepler suitable for FaaS use cases by enabling a higher frequency sampling rate

In addition, a significant facet of our involvement was assessing the accuracy of Kepler's ML model. We gauged the model's performance by comparing estimated metrics for power usage per node obtained through Kepler with real metrics gathered on Grid5000,



Mapping between PHYSICS elastic controller and KEDA components

helping to ensure the reliability of energy consumption estimates.

Autoscaling workloads

KEDA is the API for autoscaling workloads in Kubernetes clusters. With KEDA, container scaling is based on the number of events to be processed, rather than CPU or memory thresholds. In addition, it is lightweight and fully integrated with Kubernetes through CRDs (as with every PHYSICS component). KEDA works alongside the standard Kubernetes scaling components, such as Horizontal Pod Autoscaling (HPA), providing a straightforward way of extending its functionality without overwriting or duplication. It also allows managing different types of workloads—such as deployments, jobs, and even custom resources—and scaling down to zero (a plus for energy savings).

We evaluated the suitability of the scalers already present in the upstream catalog and identified a few options that we implemented and enhanced for PHYSICS/FaaS purposes. Two in particular were suitable for optimizing function wait time and function performance and providing the capability of scaling the nodes of the cluster if needed.

PHYSICS CONTRIBUTIONS

PHYSICS earned praise for its technical achievements during the conclusive project review and made substantial contributions to upstream open source communities like Kepler, Submariner, Knative, NodeRed, and more. Industrial use cases in e-health, smart agriculture, and smart manufacturing have been demonstrated in multiple publications. Those interested in learning more can explore further details on the PHYSICS website and access reusable artifacts at the PHYSICS marketplace. Source code for many components is also available on the public PHYSICS GitHub repository.

Funded by the European Union

*The **PHYSICS** project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017047.*

Join open source community conferences **DevConf.CZ** and **DevConf.US** taking place at Red Hat Research partner universities this summer!

DEVCONF.cz

Brno University of Technology
Brno, Czech Republic

June 13-15, 2024

DEVCONF.us

Boston University
Boston, USA

August 14-16, 2024

Learn more at devconf.info

RESEARCH QUARTERLY

VOLUME 6:1

Red Hat

Feature

Team threat hunting on a container platform: Kestrel as a Service

An automated tool developed by researchers aims to decrease the mean time to detection by enabling threat hunters to automate and collaborate within a secure, stable container environment.

by Kenneth Peeples

About the Author
Kenneth Peeples is a Doctor of Engineering student at Colorado State University (CSU) and a Principal Security Architect in the Red Hat Consulting division.

The automated security tools in a Security Operations Center (SOC) can handle about 80% of cybersecurity threats, leaving a substantial 20% of more sophisticated threats undetected. These are the threats likely to be most detrimental to business operations, reputation, or survival. Threat hunting is an effective, proactive method to reduce detection time and minimize the impact of a threat, but we lack tools for scaling, automating, and collaborating.

Research at Colorado State University (CSU), in collaboration with Red Hat, the Open Cybersecurity Alliance (OCA), and IBM Security, approaches this problem by developing a team threat-hunting model enhanced by automation. Our approach replaces the laborious, difficult-to-scale practice of one-person threat hunting with multiple threat hunters working largely independently in a "pack" project. It also enables the reuse (not rewriting) of existing hunting knowledge from proprietary and public hunting repositories.

DISRUPTING THE IMPACT TIMELINE

The NIST Cybersecurity Framework 2.0 defines core functions for cybersecurity outcomes at a high level. As seen in **Figure 1** (next page), Identify and Protect outcomes help prevent and prepare for cybersecurity incidents, while Detect, Respond, and Recover outcomes help discover and remediate cybersecurity incidents. Govern applies to all steps of the process. The impact timeline is critical: the longer the dwell time—the time between when an attack begins and when it is detected—the more damage is done. Sophisticated threats that elude SOC automated security tools avoid detection for up to 280 days.

We hypothesize that we can reduce dwell time by providing a team of threat hunters a container environment that is secure, scalable, persistent, and collaborative, using Kestrel, OpenC2, and associated open source projects at an enterprise scale. Using a container platform enables reusing the Kestrel container in different platforms, allows for easier project spawning, and provides management for threat-hunting teams while

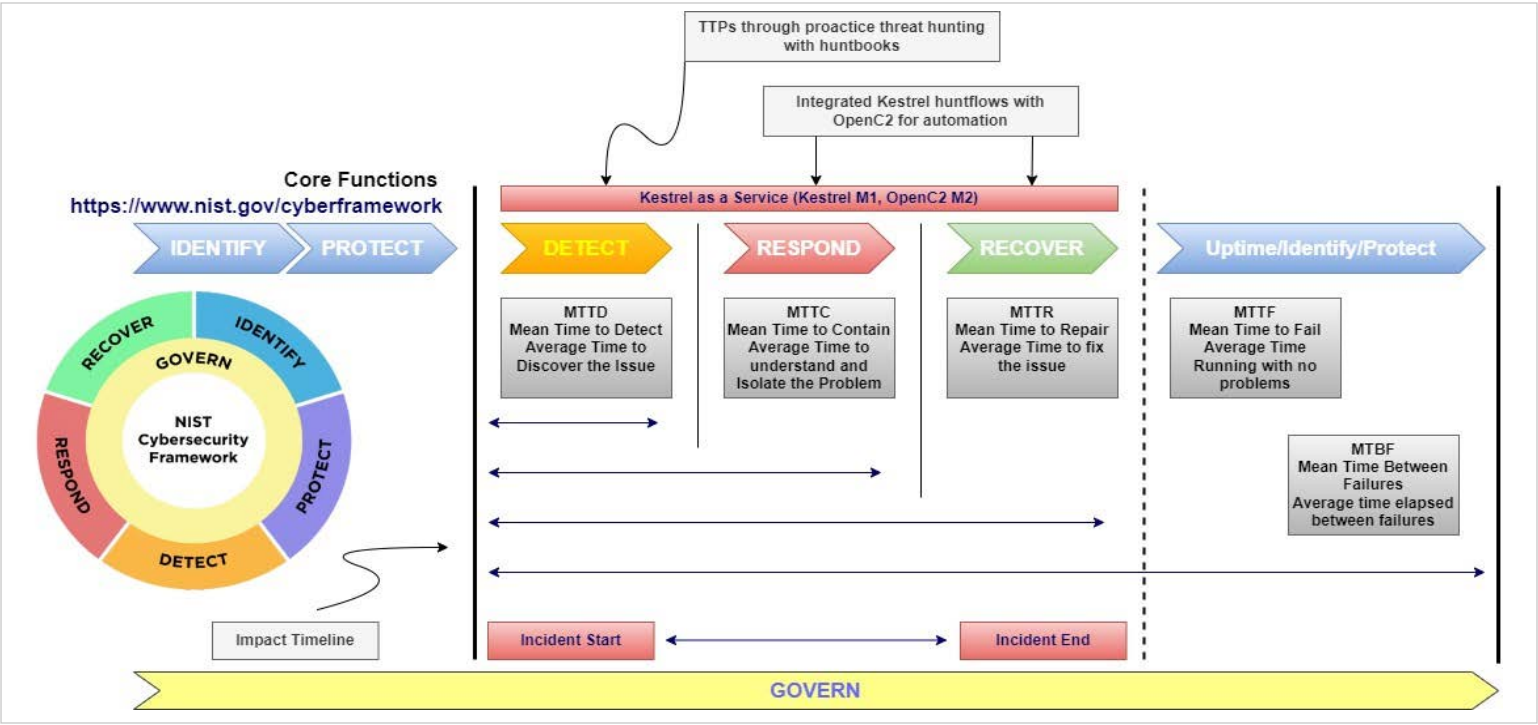


Figure 1. *Impact timeline*

providing enterprise capabilities to improve and track our metrics.

Our research focuses on a proactive hunt model based on hypothesis-based hunts that can use Indicators of Attack (IoAs) and the tactics, techniques, and procedures (TTPs) of attackers. We align our proactive hunt model to the MITRE ATT&CK framework, a globally accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is a foundation for developing specific threat models and methodologies in the private sector, government, and the cybersecurity product and service community. **Figure 2** is an example of the MITRE ATT&CK matrix for container technologies.

We want to proactively search and examine data for unnoticed security threats and use human intelligence to create hypotheses. The steps in this process can be broken down into a clear workflow.

1. Understanding the security measurements in the target environment
2. Thinking about potential threats escaping existing defenses
3. Obtaining useful observations from system and network activities
4. Developing threat hypotheses
5. Revising threat hypotheses iteratively with the previous two steps
6. Confirming new threats

If we find repeatable patterns in the data, we can use this workflow to automate some of the hunt, potentially also improving DevSecOps pipelines.

KESTREL AND CRITICAL COMPONENTS

Threat-hunting activities start with answering two questions: what to hunt and how to hunt. Any threat-hunting activity involves both types of questions, and the answers to both questions contain domain-specific knowledge. However, the domain knowledge applicable to these questions is not the same. Answers to the *what* question contain domain knowledge that is highly creative, mostly abstract, and largely reusable from one hunt to another. Answers to the *how* question guide the

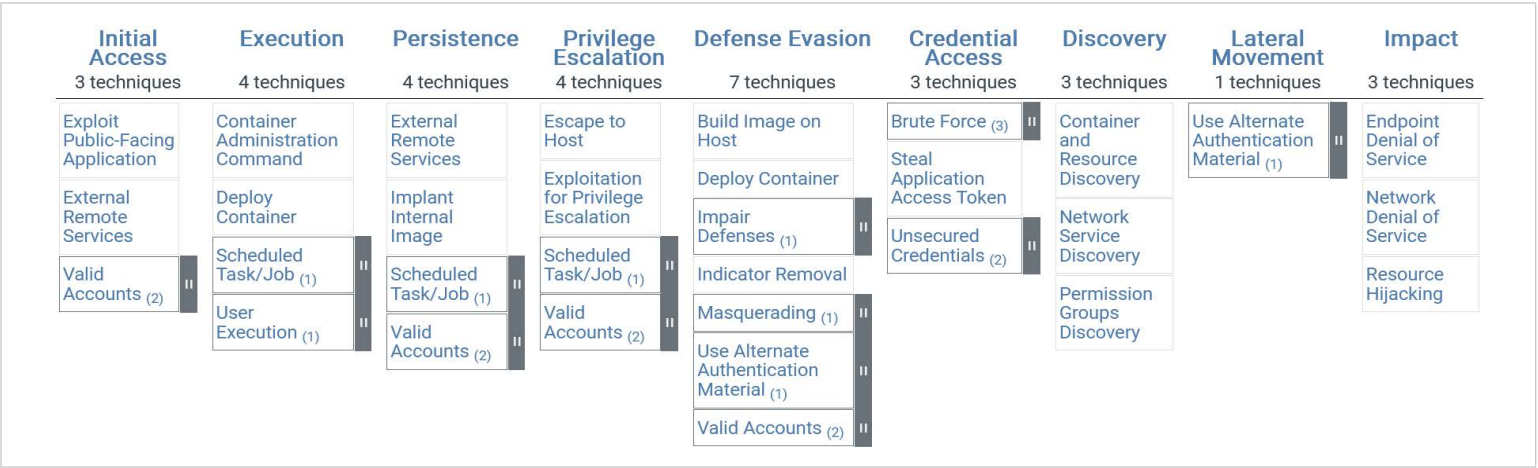
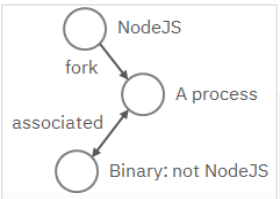


Figure 2. *MITRE ATT&CK container framework*

realization of the what and are replaced from one hunting platform to another.

These questions are both addressed by the Kestrel Project, which provides a layer of abstraction to stop the repetition involved in cyber threat hunting. Kestrel has two main components: a threat-hunting language allowing human threat hunters to express what to hunt in terms of patterns, analytics, and hunt flows, and the runtime, a machine interpreter that deals with how to hunt.

The Kestrel language offers capabilities including applying existing public and proprietary detection logic and expressing thinking across heterogeneous data and threat intelligence sources. It also allows composing reusable hunting steps, flows, and hunt books. The Kestrel runtime compiles the expression of what to hunt against specific hunting platform instructions and executes the compiled code both locally and remotely. The runtime also assembles raw logs and records into




```
exp_node = GET process FROM
stixshifter://linuxserver31

   WHERE [process:parent_ref.name = 'node' AND process:binary_ref.name
!= 'node']
   START t'2023-04-05T00:00:00Z' STOP t'2023-04-06T00:00:00Z'
```

Figure 3. *TTP Pattern and first hunt step*

human-friendly abstractions called entities (e.g., malware or Control-and-Command attack) to enable human threat hunters to create and develop threat hypotheses.

With Kestrel, we can write a pattern to match a pattern of tactics, techniques, and procedures (TTPs). For instance, one TTP pattern describes a web service exploit where a worker process of a web service, such as NGINX or NodeJS, is associated with a binary that is not the web service. This scenario is the result of an exploit of the worker process, and the common binary to execute is a shell, for example, bash.

To provide data to Kestrel, we express the TTP in a STIX pattern (**Figure 3**) using STIX-Shifter, an open source Python library that allows software to connect to products that house data repositories (e.g., SIEM systems, endpoint management systems, threat intelligence platforms, and others) by using STIX Patterning. STIX-Shifter returns results as STIX Observations. We do this so all security data, regardless of the source, looks

Block Executed In 2 seconds

VARIABLE	TYPE	#(ENTITIES)	#(RECORDS)	process*	file*	directory*	artifact*	user-account*	network-traffic*	ipv4-addr*
exp_node	process	1	133	265	314	314	133	133	11	22

*Number of related records cached.

Figure 4. *Results from the execution of the hunt step*

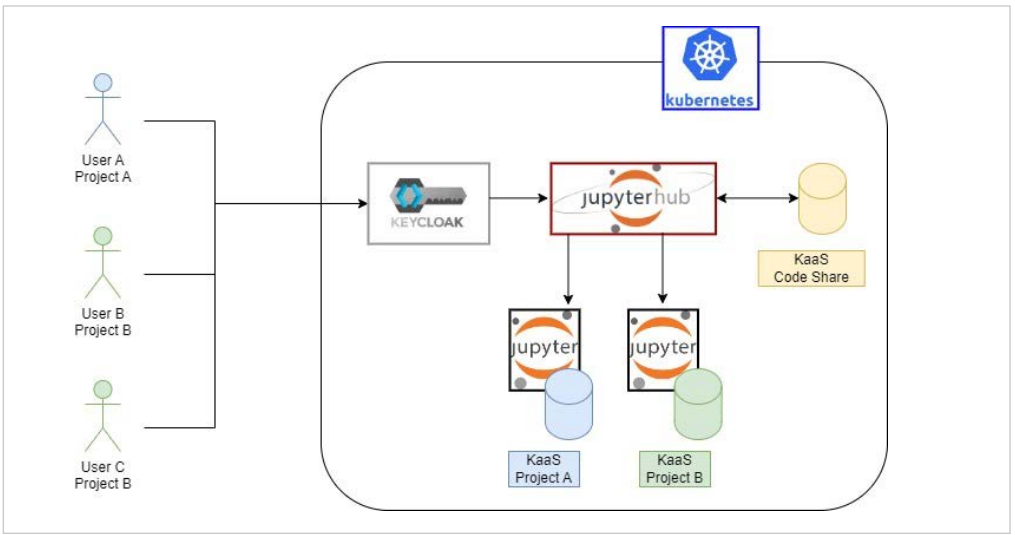


Figure 5. *A logical high-level view of the environment*

and behaves similarly. You may get results like those in **Figure 4** if there are logs that match the TTP.

The OpenC2 threat-hunting actuator profile defines the OpenC2 actions, targets, arguments, and specifiers along with conformance clauses to enable the operation of OpenC2 producers and consumers in the context of cyber threat hunting. It covers invocation of stored hunting processes (e.g., hunt books), passing of hunt parameters, selection of analytics to apply to hunt data, and the expected type(s) and format(s) of information returned by hunting processes. All of these components provide the team with a system for sharing and collaboration.

TRACKING MTTD

Our efforts focus on decreasing the Mean Time to Detect (MTTD) metric, a key measure of dwell time. We use a combination of technologies: a Kubernetes distribution plus Keycloak, JupyterHub, and a Docker container. The Docker container contains all the Kestrel components: kestrel-lang, kestrel-runtime, kestrel-analytics, and tutorials.

The container can be run standalone or with the collaboration environment; however, our goal is to realize the benefits of moving from standalone threat hunting to team threat hunting. A critical function of JupyterHub, as seen in **Figure 5**, is enabling collaboration features so that a threat-hunting team can share a Kestrel container. We are adding the capability to share the hunt steps and flows across users, projects, and organizations. Jupyterhub provides the ability to share Kestrel hunt notebooks. Keycloak provides the identity manager capability for users and roles.

Relevant metrics, including the MTTD, mean time to contain (MTTC), and mean time to repair (MTTR), are provided by the Kestrel-as-a-Service (KaaS) dashboard to track hunt project statistics. Historical metrics supplement current ones to show improvements gained by collaboration, compared to a siloed single threat hunter with tools on a local workstation who is not sharing hunt flows and steps.

The capabilities of KaaS for team threat hunting include persistence, hunt book and hunt step sharing, threat hunt project management, threat hunt pausing/restarting, and threat hunt project statistics. These capabilities reduce the time to incident detection and can be tracked through auditing and history as variables in MTTD. Average MTTD times for a single threat hunter are shown in **Figure 6**; for example, 39% of cyberattacks are detected within the span of months. Adding more threat hunters and more threat-detecting capabilities significantly reduces that span. **Figure 7** shows the impact of

additional capability from left to right on the diagram and the impact of different sizes of collaborative threat-hunting packs (2, 5, and 10 hunters).

GET INVOLVED

The next milestone for the KaaS project is finishing the threat-hunting project collaboration features, test/use cases, including GenAI, and compliance as code. We will then dive further into the comparison and analysis of the impacts of individual threat hunting with crowd hunting to determine the impact. Previous deployments were focused on development environments with Minikube, Kubernetes and Openshift AI. We are planning production deployments with users in defense, finance, as well as others.

Everyone is welcome to participate in the Open Cybersecurity Alliance. Individuals can make technical contributions to KaaS or Kestrel; OCA repositories are on GitHub. Organizations can become OCA sponsors, receive special recognition, and gain a seat on the OCA Project Governance Board. Individuals and organizations can join the Slack channel.

I encourage you to walk through the Kestrel tutorial in a Kubernetes testing environment. Instructions for deploying a KaaS development environment with Minikube can be found in the Open Cybersecurity Alliance repository on GitHub; you can then start the tutorial, located in the Kestrel documentation.

ACKNOWLEDGMENTS

I worked with several others to build Kestrel as a Service (KaaS) so teams of threat hunters can collaborate in development and enterprise environments on container platforms. This group includes Professor of System Engineering Dr. Steve Simske (CSU), Open Source Program Manager Claudia Rauch (OASIS/OCA), Security Research Scientist Dr. Xiaokui Shu (IBM), Head of Hybrid Cloud Platform Adoption Practices Stephane Lefrere (Red Hat), along with others. The project started with the guides and technologies to build and deploy to a development environment, Minikube, and production environment running on a Kubernetes Cluster.

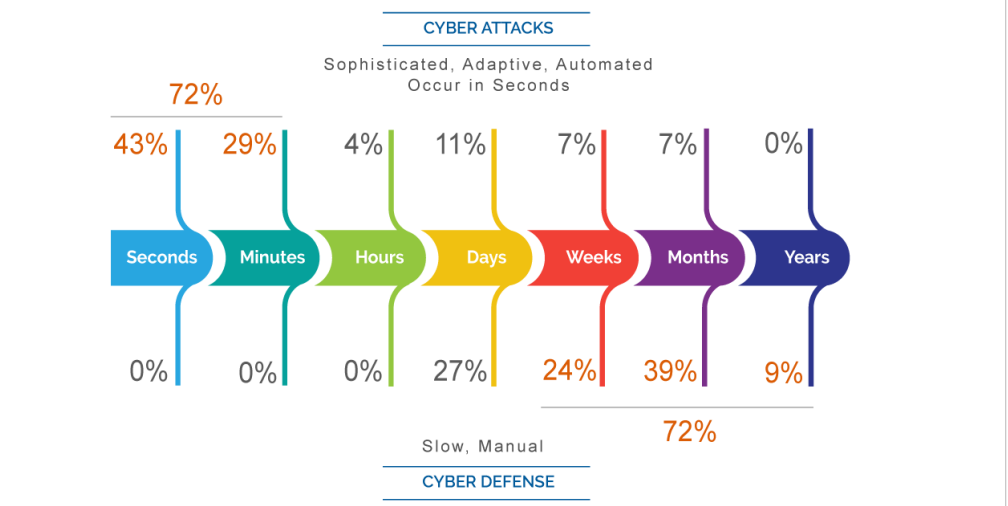


Figure 6. *Detection time vs attack time (chart from Open C2)*

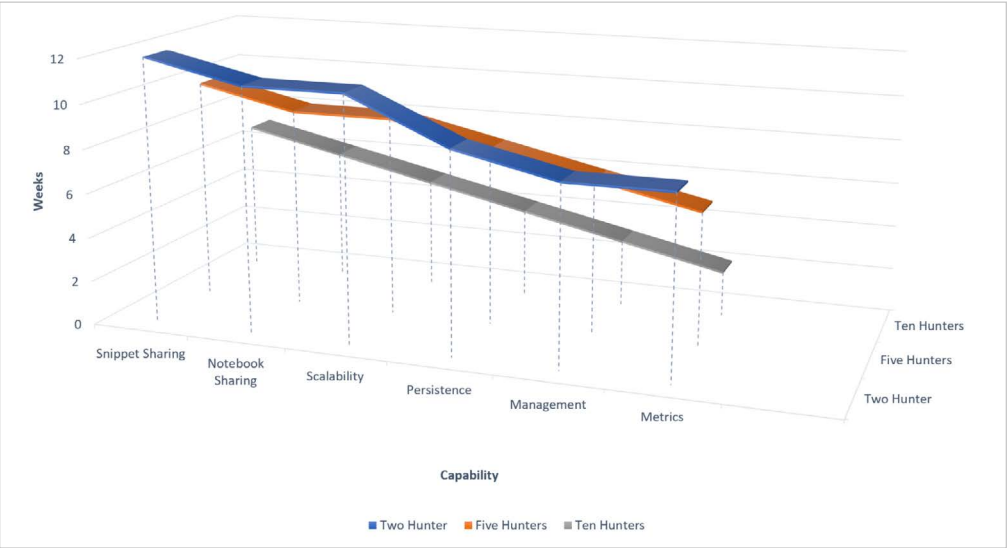


Figure 7. *Estimated impact on MTTD according to team size and capabilities*

UMass Lowell is proud to collaborate with Red Hat, a Select Preferred Partner, and celebrate more than a decade of working together on research, philanthropy and building the next generation of Red Hat's workforce.

UMASS LOWELL

RESEARCH QUARTERLY

VOLUME 6:1

Red Hat

Column

Focus on trust

Elements of trust are nearly ubiquitous in software development, spanning from security concerns to trustworthiness and reliability. Current projects address the question of trust in many aspects.

by Martin Ukrop

About the Author
Martin Ukrop
is a Principal Research Software Engineer with Red Hat Research, focusing on security research and facilitating industry–academia cooperation in EMEA. He received his PhD in Computer and Information Systems Security from Masaryk University, Czechia, focusing on human aspects of computer security. He remains an active teacher as well as a life-long learner.

Red Hat Research and its university partners focus strategically on projects with the most promise to shape the future of how we use technology. Each quarter, RHRQ will publish an overview of our research in a specific area, such as edge computing, hybrid cloud, and security. In this issue, we focus on projects related to trust.

Several projects at Red Hat Research pertain to the notion of trust. There are multiple views of trust, however: it naturally arises in security (trust put by users in data confidentiality, integrity, and provenance) but manifests itself in quality assurance as well (users expecting reliable, bug-free, trustworthy software).

There seems to be a huge gap in the level and maturity of these aspects of trust in the wild. While most large companies heavily invest in reliability and security across their portfolios, many organizations are only at the beginning of their journey to make their products secure and trustworthy.

Although the market is full of trust-related tools addressing individual issues, these are just small, discrete blocks. Building a comprehensive, trustworthy solution from them is usually rather challenging, and the result is often fragile. Although to build a full solution, one must start by creating the product or service blocks, it's essential to also bind them into a coherent end-to-end design for deployment once all the blocks are ready. This is especially critical because the weakest link determines the overall trust level.

Below, we'll look at the midterm outlook for five trust-related research collaborations between Red Hat Research and universities in Boston, Massachusetts, and Brno, Czechia. While some projects represent very specialized blocks addressing a particular trust issue, others attempt to integrate multiple pieces and span a wider part of the ecosystem.

SECURITY

The Sec-certs project, a cooperation with Masaryk University, Czechia, looks at the ecosystem

of security certifications (e.g., Common Criteria, FIPS 140. FedRAMP). By analyzing the available metadata, parsing the available PDF documentation for each certification, and cross-referencing other datasets (e.g., CPEs, CVEs, CWEs), Sec-certs attempts to piece together multiple existing blocks in the domain. The outlook for 2024 includes creating a dashboard showing ecosystem statistics in real time, stabilizing the codebase, and showcasing the tool to the community at multiple global certifications events. Another open direction is involving sophisticated natural language processing tools to enrich the dataset even more.

Two projects in the Red Hat Collaboratory at Boston University consider the security aspects of trust. The CoFHE project prepares for a future with seamless encrypted computing in the cloud by building a compiler to make fully homomorphic encryption (FHE) more accessible for use cases like data science. Although speeds of FHE move into the feasible range when using hardware acceleration, designing the corresponding code still remains a specialized task for security engineers. CoFHE proposes a comprehensive FHE compiler framework to automate the process of generating implementations using the Cheon-Kim-Kim-Sing (CKKS) scheme. It targets machine learning applications due to their current pervasiveness in the cloud. During 2024, the authors plan to do both the necessary background modeling work and develop an initial design for the compiler framework.

The second Collaboratory project, the HySe project investigates hypervisor security through component-wise fuzzing. Considering the complicated building structure of today's hypervisors, HySe proposes to identify less usual interfaces between the hypervisor and its guest VMs (e.g., VM migration interface of disk modification interface). By applying fuzzing techniques even in these less exposed places, HySe will help strengthen the crucial isolation guarantees the hypervisor should provide. Throughout 2024, the project team plans to, first, identify existing interfaces and conduct the appropriate threat modeling on them. Second, they will design, implement, and evaluate program analysis techniques to preemptively identify bugs and vulnerabilities in the individual hypervisor components that form the exposed interface surface.

QUALITY ASSURANCE

The Lock 'n Load project, also a part of the Red Hat Collaboratory, sets out to address an underrated specialized issue: the inability to detect deadlocks in binary-only kernel modules. For cases where source code is available, an existing Lockdep tool can be used for this task. However, Lockdep's detection mechanisms fall short for binary-only kernel modules such as proprietary drivers. Over 2024, the project aims to decouple metadata from locking data structures, automatically configure and build a suitable kernel, and evaluate Lock 'n Load's deadlock detection ability.

Last but not least, Project Perun, a cooperation with Brno University of Technology, Czechia, combines multiple blocks of software development and quality engineering to increase software trustworthiness, aiming to help reliably identify root causes of performance degradations. Perun binds code performance profiles to the project's version control system, thus allowing QE engineers to identify offending commits and functions quickly. Since early 2024, the researchers have been intensely cooperating with the kernel performance team at Red Hat to pilot test the tool in reliability and performance testing of kernel versions for an upcoming RHEL 10 release.

Explore these and other security-related projects in the research directory of the Red Hat Research website.

See "Preserving privacy in the cloud: speeding up homomorphic encryption with custom hardware," RHRQ 4:3 (Nov 2022).

THE UNIVERSAL AI SYSTEM FOR HIGHER EDUCATION AND RESEARCH

NVIDIA DGX A100

Higher education and research institutions are the pioneers of innovation, entrusted to train future academics, faculty, and researchers on emerging technologies like AI, data analytics, scientific simulation, and visualization. These technologies require powerful compute infrastructure, enabling the fastest time to scientific exploration and insights. NVIDIA® DGX™ A100 unifies all workloads with top performance, simplifies infrastructure deployment, delivers cost savings, and equips the next generation with a powerful, state-of-the art GPU infrastructure.

Learn More About **DGX** @ nvda.ws/dgx-pod
Learn More About **DGX on OpenShift** @ nvda.ws/dgx-openshift

GET A LAPTOP THAT IS AI READY

AMD RYZEN™ AI TECHNOLOGY IS NOW BUILT IN

AMD
RYZEN AI

AMD RYZEN 7000 SERIES
AMD RADEON GRAPHICS

*Available on selected systems