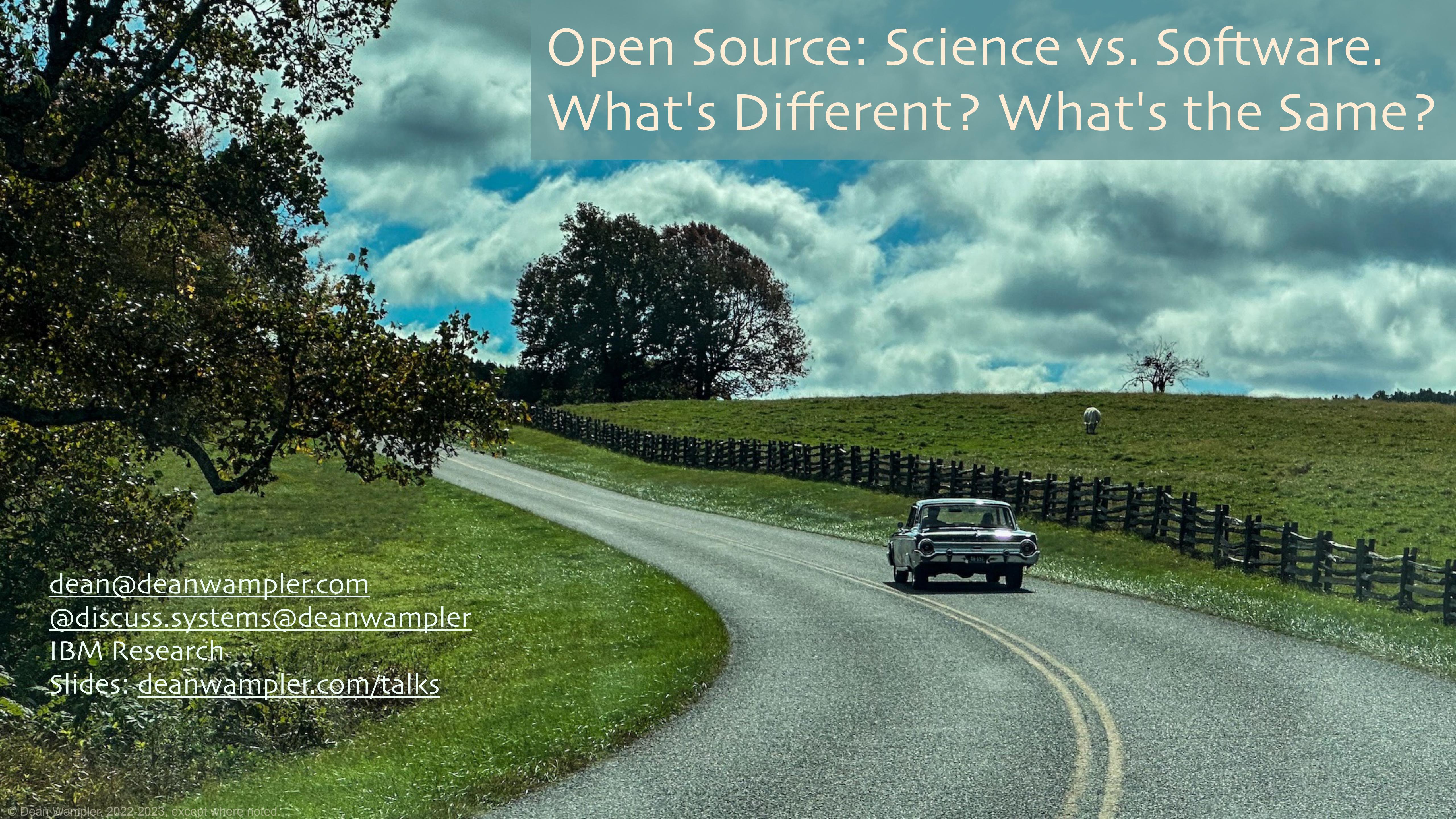


Open Source: Science vs. Software. What's Different? What's the Same?

A scenic rural road curves through a green landscape under a dramatic, cloudy sky. A classic sedan drives away from the viewer on the road. A large, leafy tree stands prominently on the left side of the road. In the background, rolling hills are visible, separated by a wooden fence.

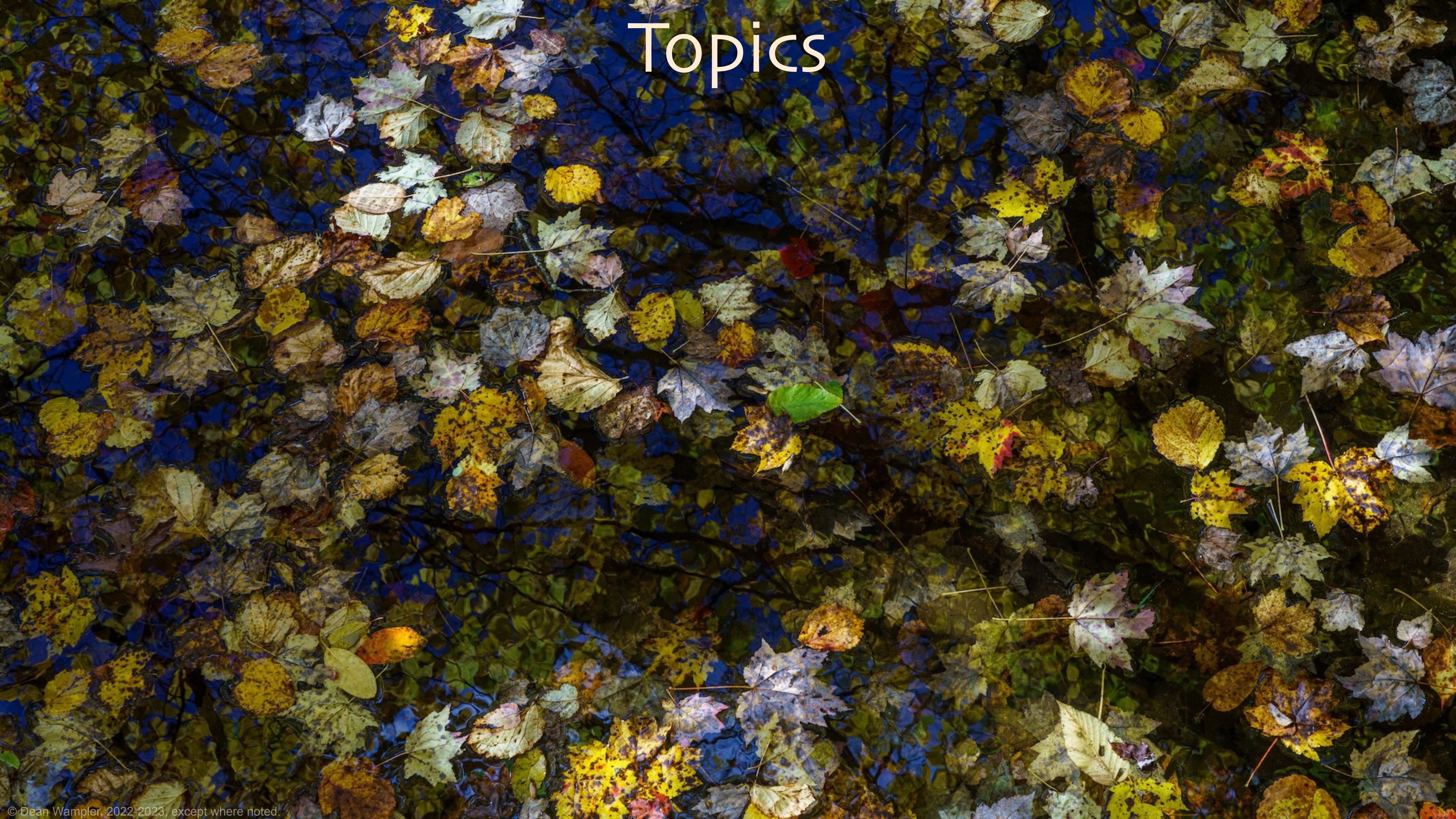
dean@deanwampler.com

@discuss.systems@deanwampler

IBM Research

Slides: deanwampler.com/talks

Topics

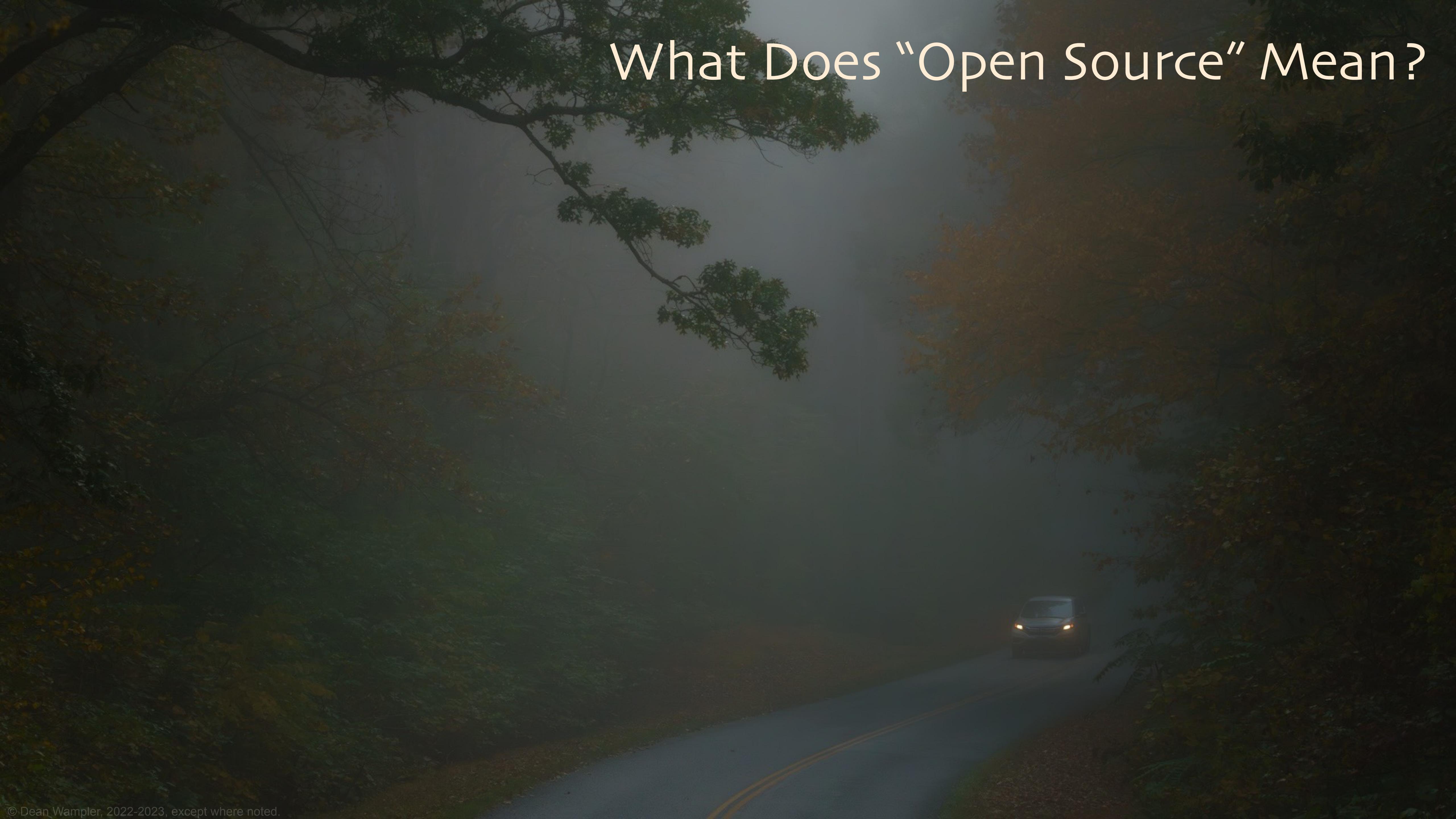


Topics

- What Does “Open Source” Mean?
- The Different Motivations and Goals for OSS and OSSci
- What Can OSSci Learn from OSS?
- What Can OSS Learn from OSSci?

- OSS: Open Source Software
- OSSci: Open Source Science

What Does “Open Source” Mean?



What Does “Open Source” Mean?

- Characteristics according to the Open Source Initiative:
 - Free redistribution
 - Includes the source code
 - No discrimination - people, institutions, or applications
 - Derived works permitted under the same license
 - License, automatic distribution with software, not product-specific, restrictive of other software, ...

What Does “Open Source” Mean?

- But opinions and situations differ:
 - Free redistribution
 - But what about commercial vs. non-commercial use?
 - Includes the source code
 - But what about data?
 - No discrimination - people, institutions, or applications
 - But what about use in weapons, nuclear plants (e.g., Java...)?
 - ...

What Does “Open Source” Mean?

- But opinions and situations differ:
 - ...
 - Derived works permitted under the same license
 - Gnu General License limits use in proprietary software
 - License, ...
 - A major headache is all the license choices.
 - Advice: Avoid GPL. Use Apache, MIT, or Berkeley licenses.

A Little History (Wikipedia)

- Automobile patents shared in the early 20th century.
- IBM Mainframe software distributed as source
 - SHARE (not an acronym) and GUIDE (an acronym)
- BSD, then eventually Linux

A Little History (Wikipedia)

- The term “Open Source”:
 - “Free Software” movement
 - “Open Source” suggested by Christine Peterson
 - Netscape Navigator
 - Eric S. Raymond, et al. started the Open Source Initiative
 - Tim O'Reilly, Open Source Summit (today's version)

The Different Motivations and Goals for OSS and OSSci



The Different Motivations and Goals for OSS and OSSci

- Software Engineers (SWEs) want to:
 - Collaborate with industry peers on common needs
 - Saves effort
 - Yields the best quality
 - Spreads learning
 - Provides personal rewards
 - Achievement recognition
 - Meets desire to interact with peers who share common interests

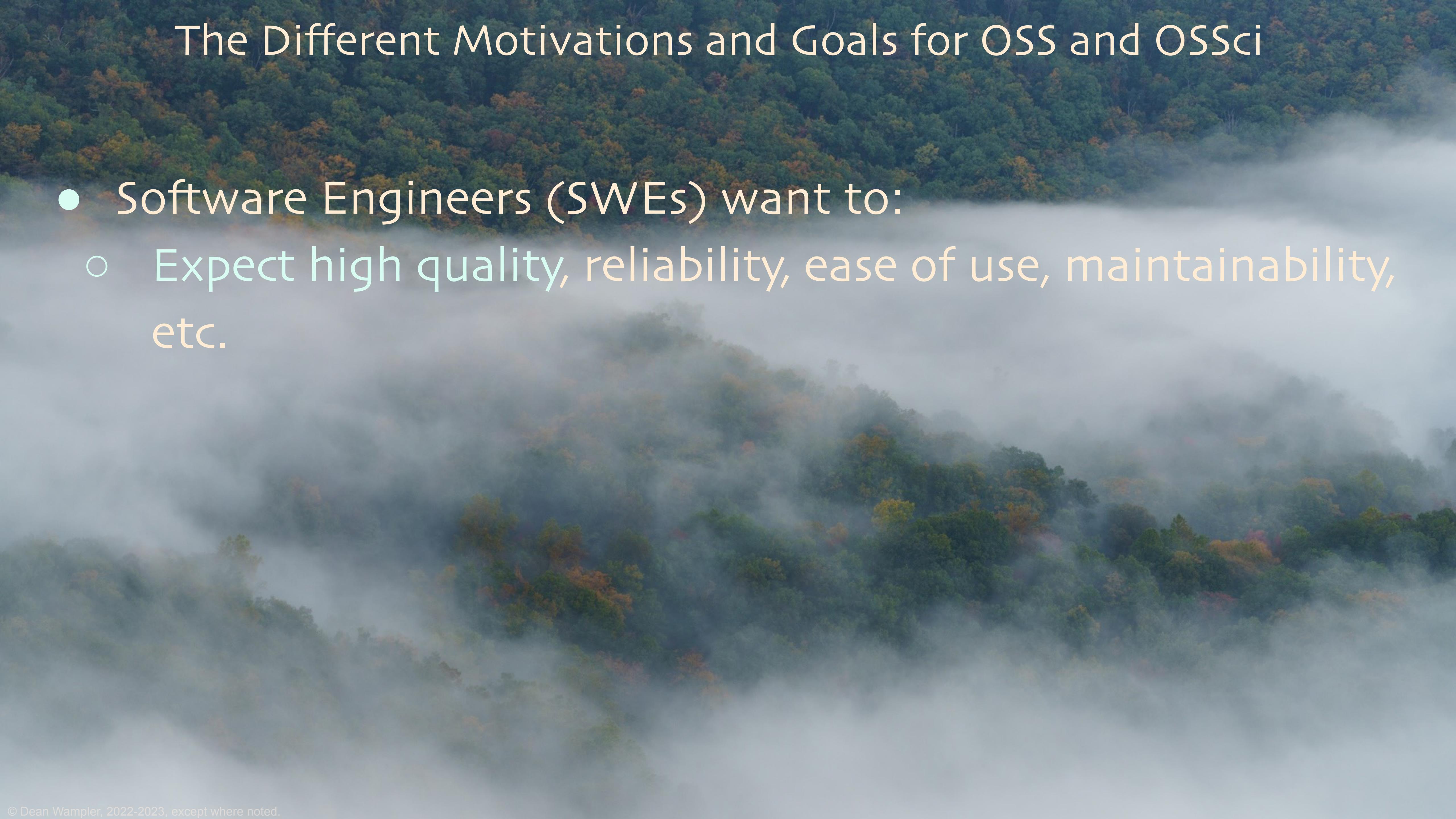
The Different Motivations and Goals for OSS and OSSci

- Software Engineers (SWEs) want to:
 - Evaluate an OSS project:
 - Clone an OSS repo and determine what it does, is it useful, and is the quality sufficiently high
 - Therefore building, installing, and adopting the software has to be as easy and automated as possible

- Clone: Make a local copy of a repo
- Repo: “Asset” repository that tracks versions

The Different Motivations and Goals for OSS and OSSci

- Software Engineers (SWEs) want to:
 - Expect high quality, reliability, ease of use, maintainability, etc.



The Different Motivations and Goals for OSS and OSSci

- Scientists^{*} want to:
 - Share ideas with colleagues
 - Enable reproducibility of results

- Scientists^{*}: Used for brevity. Also research SWEs and other staff.

The Different Motivations and Goals for OSS and OSSci

- Scientists usually don't need to:
 - Use their software in “production” scenarios
 - ... but not always!

- Production: Catch-all term for deploying software for others to use... and rely on.



The Different Motivations and Goals for OSS and OSSci

- OSSci is more likely to include data.
- OSS is less likely to include data.
 - But open AI models, which are data and code, and the training data sets used to create them are a very important topic right now!

What Can OSSci Learn from OSS?



What Can OSSci Learn from OSS?

- OSS Excels at Reproducibility
 - The relatively new requirement for research reproducibility can be achieved in part with tools long familiar to SWEs.
 - You should learn and use the most important techniques they use...

What Can OSSci Learn from OSS?

- Asset management in git repos
 - Share work with collaborators
 - Enable concurrent work with branches
 - Track all changes
 - Analogous to using a lab notebook to record what you did, what worked (or didn't)
 - Know what is in every release
- Asset: code, docs, notebooks, build and deployment tools, and data
- Git: The dominant tool for this purpose, especially GitHub
- Branch: different sequence of modifications. Branches can be merged when desired
- Release: a snapshot of the repo's "main" branch that can be shared and used by others

What Can OSSci Learn from OSS?

- Verify behavior
 - Use tests to confirm the software behaves as expected
 - Rigorous and thorough software test processes exist,
 - but they aren't used as much as they should be used.
 - Still, OSS tends to follow better practices than proprietary software projects

What Can OSSci Learn from OSS?

- Automation
 - Tests (and build and installation) processes are great,
 - but tedious and error prone to do by hand
 - Make all processes automated, on-demand, fast. and robust

What Can OSSci Learn from OSS?

- Language ecosystem
 - Pick the language ecosystem that works best...
 - for your domain,
 - for the problems that need solving,
 - for the skill level of the team

- Language ecosystem: Catch-all term for the programming language, libraries, tools for running the code, etc.

What Can OSSci Learn from OSS?

- Things I've seen scientists overlook
 - Managing dependencies
 - Having too many dependent libraries and tools, especially when you require two or more versions of the same library!
 - Makes your software far less useful

What Can OSSci Learn from OSS?

- Things I've seen scientists overlook
 - CVEs ("Common Vulnerabilities and Exposures")
 - i.e., security holes, risks to you and your users
 - Keep dependencies up to date; they "patch" CVEs.

What Can OSS Learn from OSSci?



What Can OSS Learn from OSSci?

- Open Source Code vs. Open Source Data
 - Data has been important for reproducible science for a long time
 - It is sometimes more important than the research code used to analyze it
 - Not previously true for OSS, but suddenly really important for AI

What Can OSS Learn from OSSci?

- Deterministic vs. Probabilistic and Statistical Results
 - Science is accustomed to working with P&S results
 - Experiments are never deterministic, even in the classical world (i.e., no Quantum effects)
 - AI and ML before it have forced SWEs learn and use P&S
 - Lots of active research topics in Generative AI, too

What Can OSS Learn from OSSci?

- Scale-out Distributed Computing, e.g., HPC
 - Science has long needed cluster-scale computing for:
 - Data processing from giant experiments, like at Fermilab and CERN
 - Simulations of complex fluid flows, QCD,
 - Others...
 - Science pioneered HPC for cluster computing, which predate AWS, Hadoop, Kubernetes, etc.
 - Even specialized hardware

• HPC: High performance computing

Summary

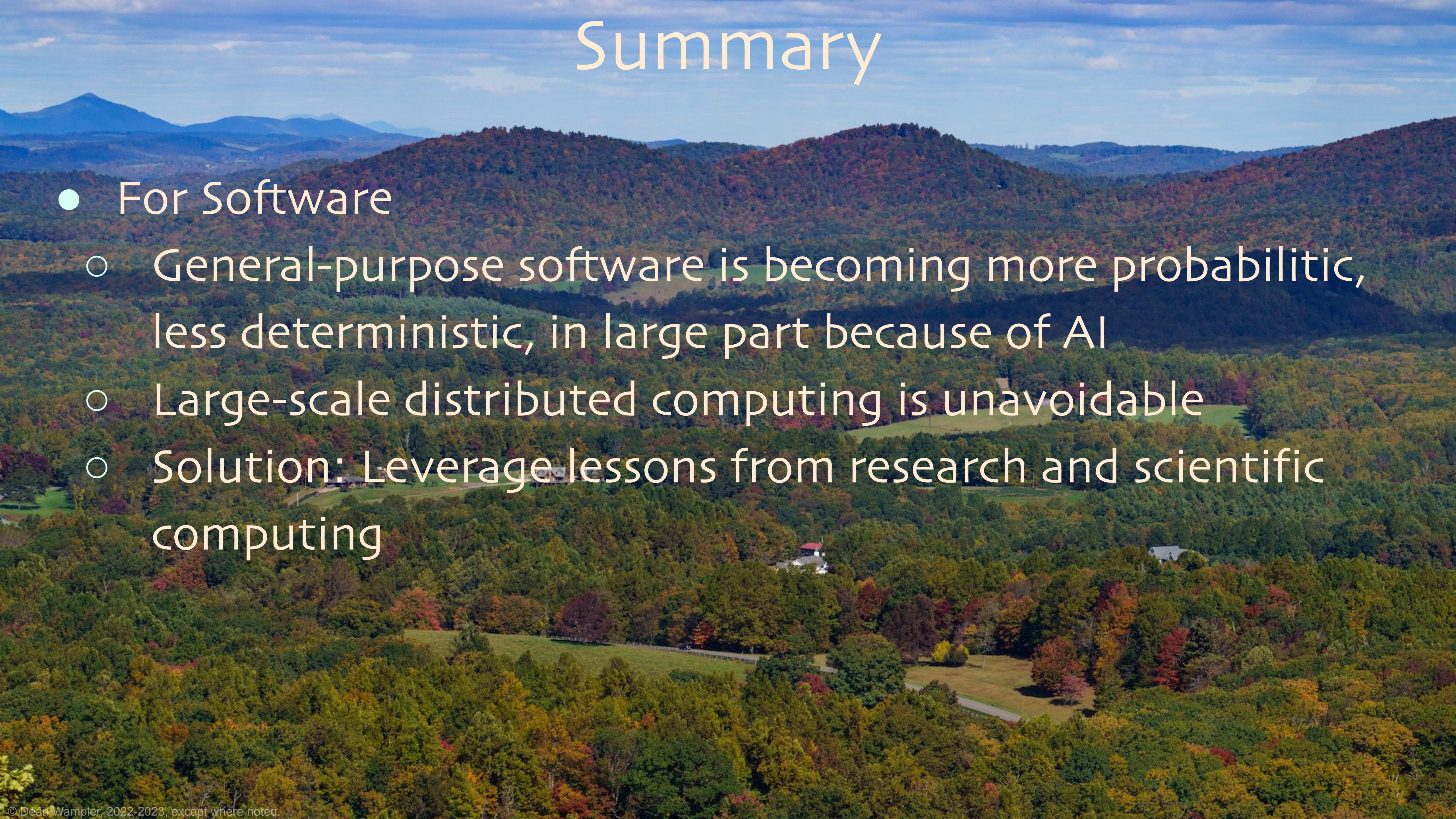


Summary

- For science:
 - More and more, scientific research needs quality software
 - Reproducibility of results and collaboration with peers are essential
 - Solution: OSSci and leverage OSS software engineering tools and practices

Summary

- For Software
 - General-purpose software is becoming more probabilistic, less deterministic, in large part because of AI
 - Large-scale distributed computing is unavoidable
 - Solution: Leverage lessons from research and scientific computing





Questions?

dean@deanwampler.com
@discuss.systems@deanwampler
IBM Research
Slides: deanwampler.com/talks