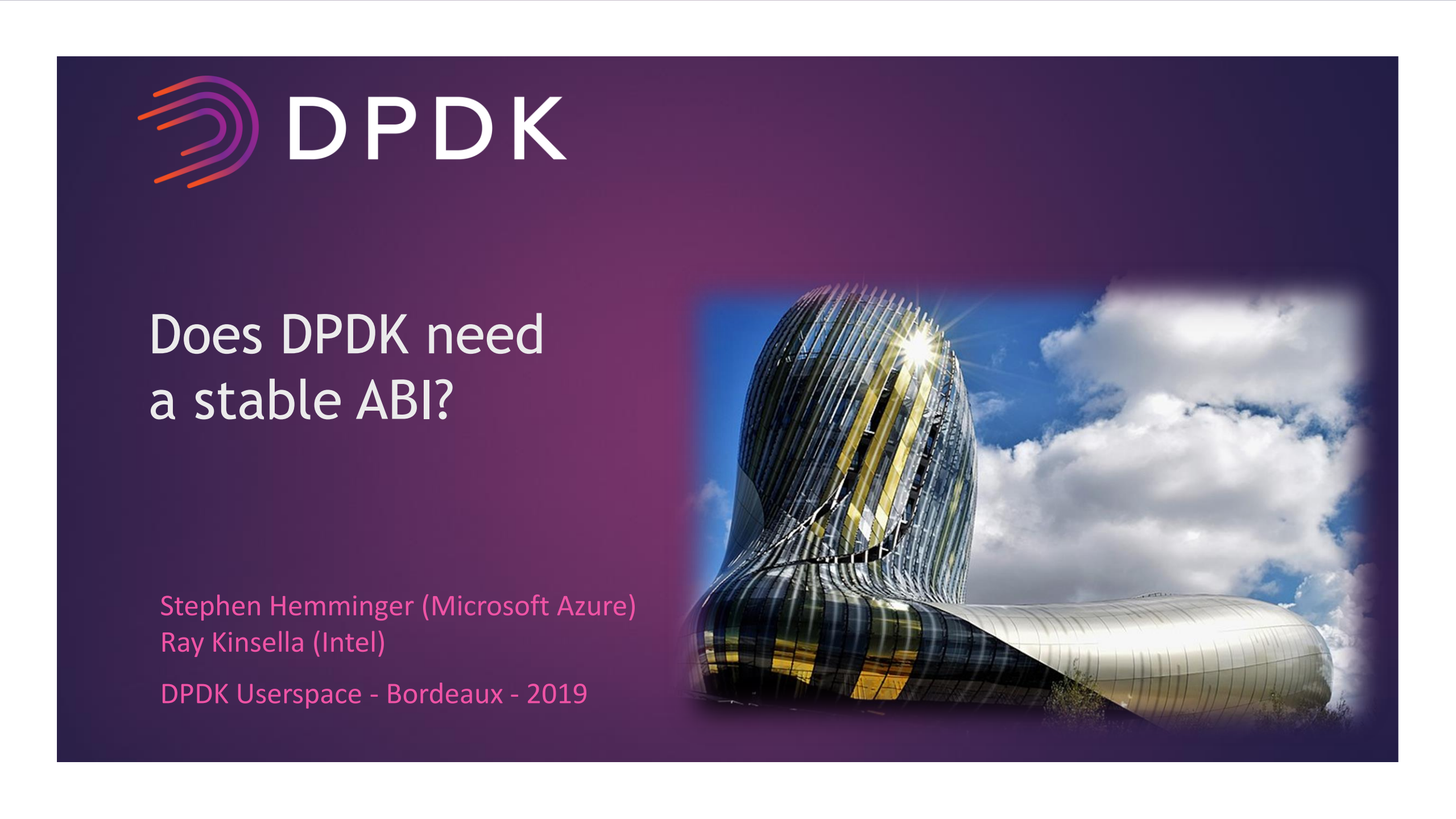


Does DPDK need a stable ABI?

Stephen Hemminger (Microsoft Azure)
Ray Kinsella (Intel)

DPDK Userspace - Bordeaux - 2019

Legal Notices & Disclaimers

- This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.
- Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.
- Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.
- The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
- Intel, the Intel logo, Atom, Xeon, Xeon Phi, 3D Xpoint, Iris Pro and others are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Adapting to change?

Development Processes

Waterfall → Agile → DevOps

Application Architecture

Monolithic → Multi-tiered → Micro-Services

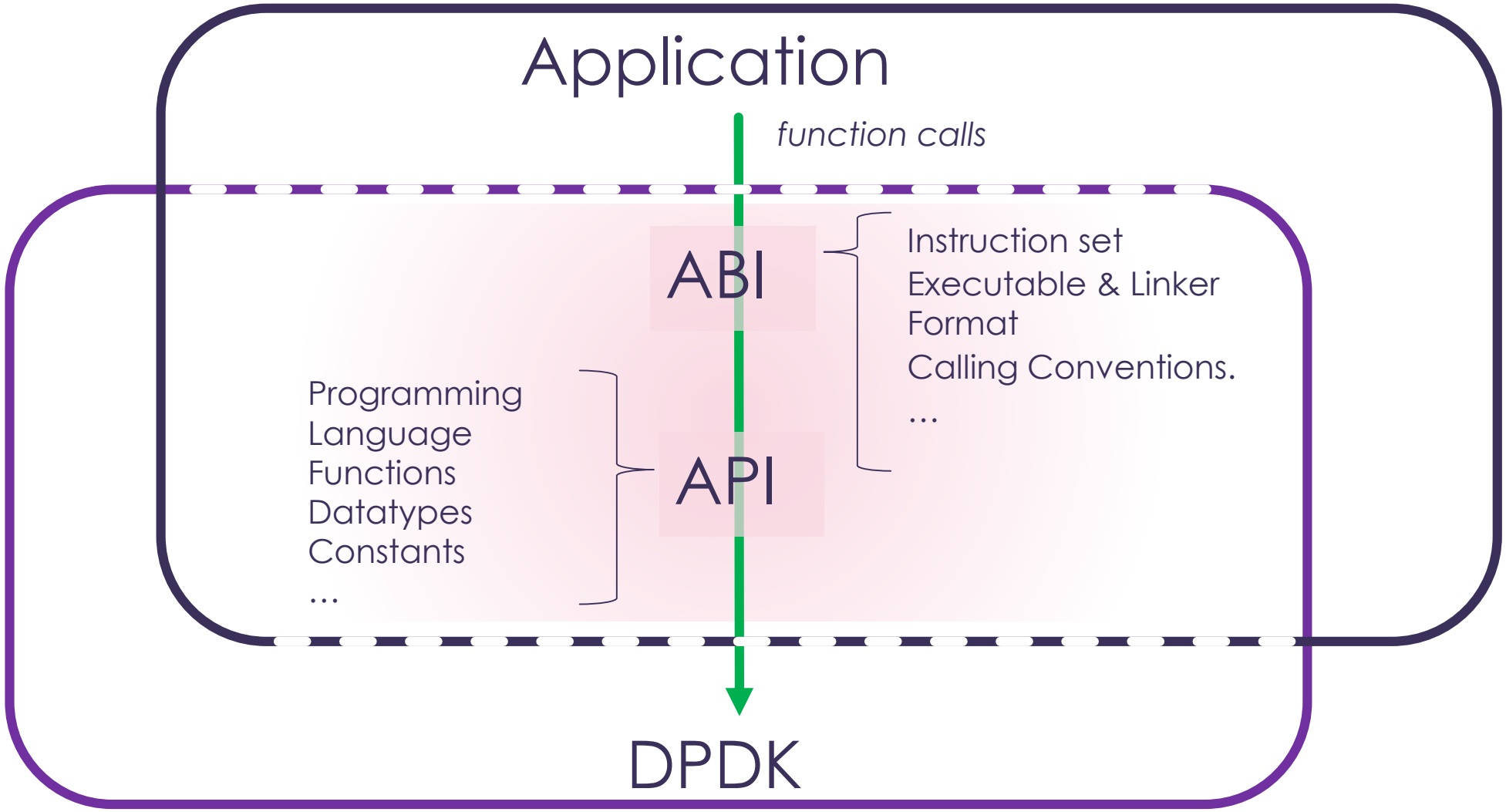
Deployments

Appliances → Virtual Machines → Containers

Data-Plane Development Kit

Proof of Concept → Open Source Project → Product

What is an ABI?



Why ABI Stability?

Maintaining a stable ABI for this change is impossible!

Contributors

There is too much DPDK ABI churn, keeping up to date is impossible!

Application Developers

These network apps are just broken and it is impossible to update them!

NetOps Engineers

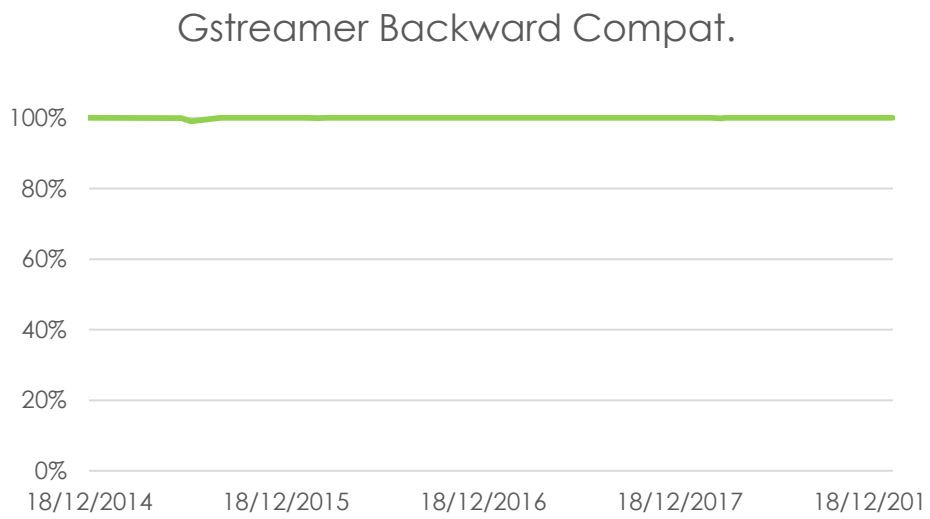
Linux Kernel ABI Stability

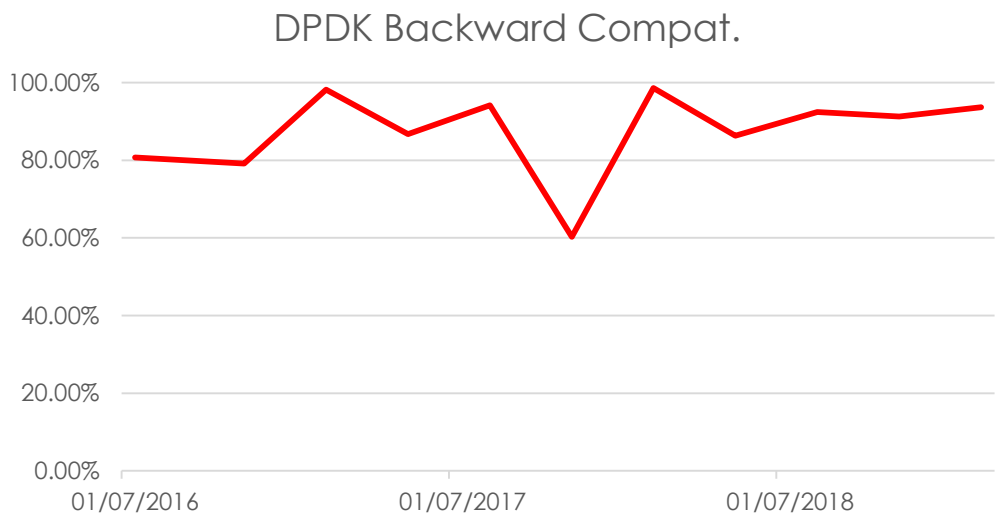
We care about user-space interfaces to an insane degree. We go to extreme lengths to maintain even badly designed or unintentional interfaces. Breaking user programs simply isn't acceptable.

Linus Torvalds, 2005

Linux Tux, by Larry Ewing 1996.
Public Domain

Comparative ABI Churn





GStreamer Application Binary Interface

- 100% backward compatible within Major Versions (1.x).
- Stable since 1.4.5, typically < 1% change between Major Versions.

https://abi-laboratory.pro/index.php?view=timeline&l=gstreamer

DPDK Application Binary Interface

- 8.7% median ABI churn between quarterly releases.
- LTS release ***is*** API stable for 2 years, however limited backporting of new features or HW.

https://abi-laboratory.pro/index.php?view=timeline&l=dpdk

Thanks to ABI Laboratory for providing the ABI analysis tools.

Proposals on ABI Stability

- Major ABI versions are declared every **year** and are then supported for one year, typically aligned with the LTS release.
- The ABI version is managed at a project level in DPDK, with the ABI version reflected in all library's soname.
- The ABI should be preserved and not changed lightly. ABI changes must follow the outlined deprecation process.
- The addition of symbols is generally not problematic. The modification of symbols is managed with ABI Versioning.
- The removal of symbols is considered an ABI breakage, once approved these will form part of the next ABI version.
- Libraries or APIs marked as Experimental are not considered part of an ABI version and may change without constraint.
- Updates to the minimum hardware requirements, which drop support for hardware which was previously supported, should be treated as an ABI change.

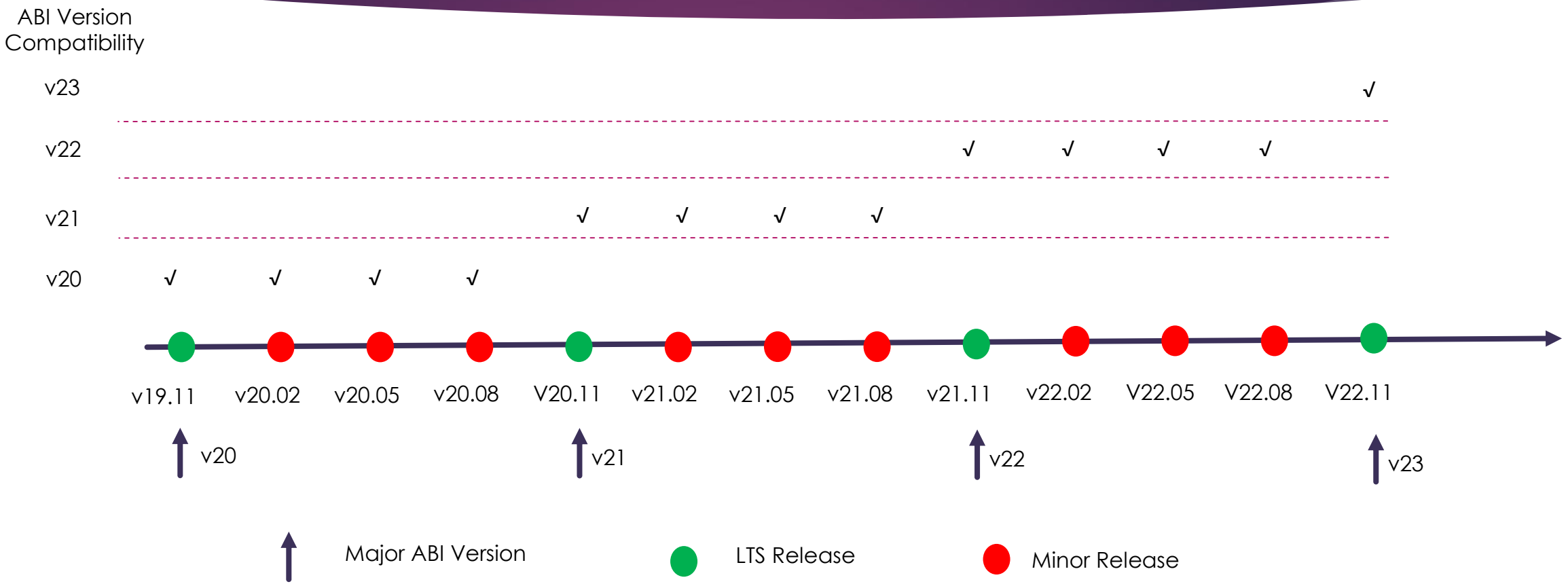
ABI / API remains stable for one year

New platform, features & improvement continue to be released.

Experimental libraries or APIs may change without restriction.

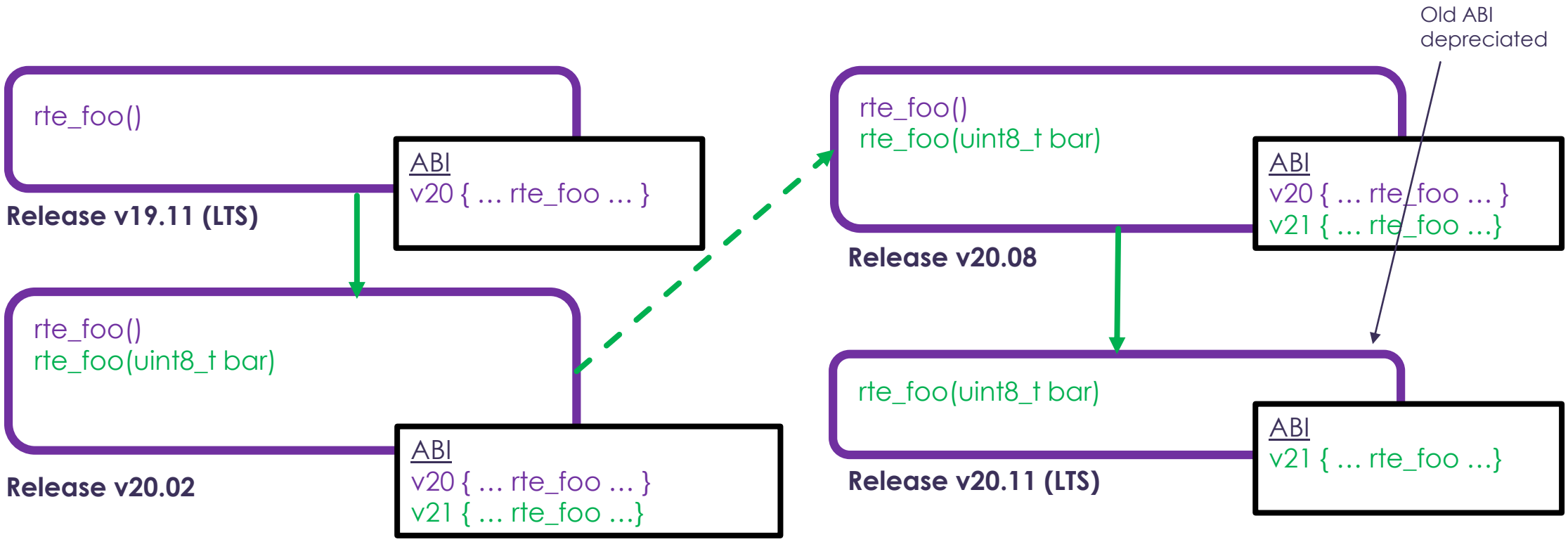
[dpdk-dev] [PATCH v3 0/4] doc: changes to abi policy introducing major abi versions

ABI Stability: major ABI versions



- Major ABI versions are declared every year aligned with the LTS release.
- ABI / API remains stable for one year, from the LTS.

ABI Stability: addition and modification



- The addition of symbols is generally not problematic. The modification of symbols should be managed with ABI versioning.
- New platform, features & improvements continue to be released.

Where the ABI policy applies

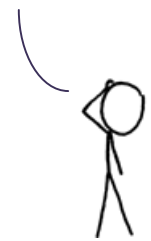
In Scope	Out of Scope
• lib • drivers (with public API's)	• lib (experimental) • api (annotated experimental) • examples • usertools • devtools • doc

Our choices …

	Best Case	Worst Case
What if we do not make this change?	Application developers are slow to adopt new versions.	Application developers start looking for an alternative.
What if we do make this change?	Contributors adding new features, need to take care of ABI compatibility	Changes breaking the ABI, get pushed to the next major ABI version

Integrating new DPDK versions is soooooooo painful ….

Gonna take some extra time, need to ensure ABI compatibility….



Next Steps

- Tentative agreement at the DPDK Technical Board.
- Supporting work - preparation for ABI stability
  - Review of exposed data structures and inline functions.
  - Supporting infrastructure to test ABI stability in the UNH lab.
  - Ensure testing of ABI Compatibility as part of the RC Test Cycle.
  - Removal of old ABI Versioning code.

Pending today's feedback, now is the opportunity to voice how this change affects you!

Questions?

Stephen Hemminger

stephen@networkplumber.org