



# PACKAGING AND VERSIONING

Ben Walker

Data Center Group

Intel Corporation

# NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at [intel.com](http://intel.com), or from the OEM or retailer.

No computer system can be absolutely secure.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit <http://www.intel.com/performance>.

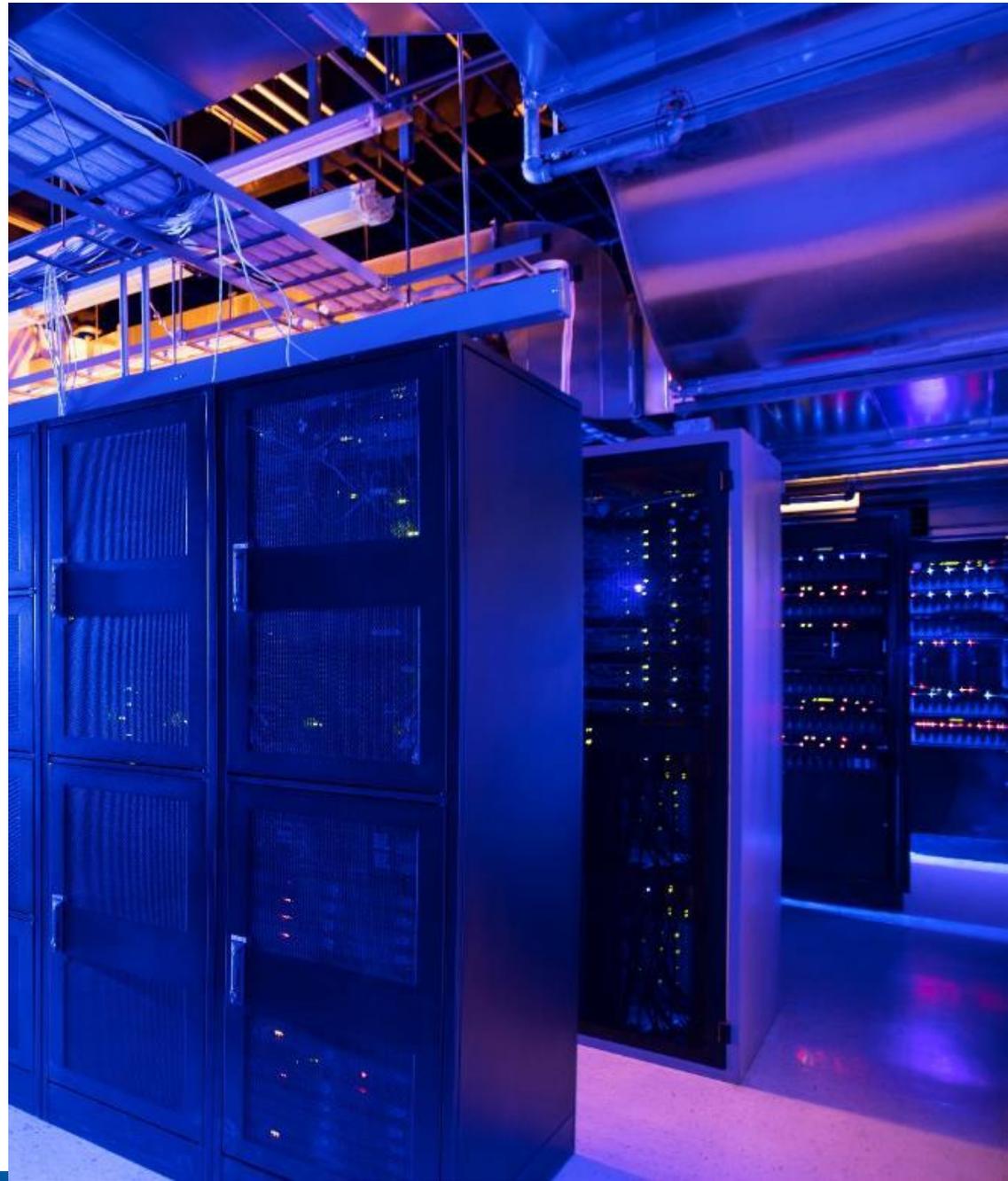
Intel, the Intel logo, Xeon, and others are trademarks of Intel Corporation in the U.S. and/or other countries.

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

© 2017 Intel Corporation.

# AGENDA

- INTRODUCTION
- API COMPATIBILITY
- ABI COMPATIBILITY
- PACKAGING



# INTRODUCTION

**REAL PRODUCTS ARE SHIPPING WITH SPDK**

**WE NEED A STRATEGY FOR DEALING WITH BACKWARD COMPATIBILITY**

# WHAT DO WE HAVE TODAY?

- 4 Time-based Releases a Year
- We will back-port bug fixes to releases by request.
- We will also accept back-port patches
- `<spdk/version.h>` contains pre-processor macros to programmatically detect SPDK version



# API COMPATIBILITY PROPOSAL

# DEFINITIONS

- **Stable**

- When a header is marked as stable, significant effort will be invested to avoid breaking the API going forward.
- APIs that will be broken will become deprecated for a full release prior to being modified.

- **Unstable**

- No special effort is made to maintain compatibility.



# STABLE HEADER CANDIDATES

## Likely Candidates

- nvme.h
- env.h
- io\_channel.h
- log.h
- nvme\_spec.h
- nvme\_spec.h
- Lots of the minor headers

## Unlikely Candidates

- nvme.h
- bdev.h
- scsi.h
- vhost.h
- blob.h

# ABI COMPATIBILITY PROPOSAL



# PACKAGING PROPOSAL

# COMPONENT OVERVIEW

## Applications

- nvmf\_tgt
- lscsi\_tgt
- Vhost\_tgt
- Perf
- Fio\_plugin
- spdkcli

## Libraries

- libspdk\_nvme.a
- Libspdk\_nvmf.a
- Libspdk\_vhost.a
- Libspdk\_blob.a
- Etc.

# PACKAGING THE LIBRARIES

## Option 1: Package static libraries

- Pro: You only link in what you need
- Con: You have to know what libraries you need
- Con: No future chance of ABI compatible upgrades

## Option 2: Package a shared library

- Con: You link in the whole thing
- Pro: It's easy to use; just add `-lspdk`
- Pro: Sets the project up for future ABI compatible upgrades

# PACKAGING THE APPLICATIONS

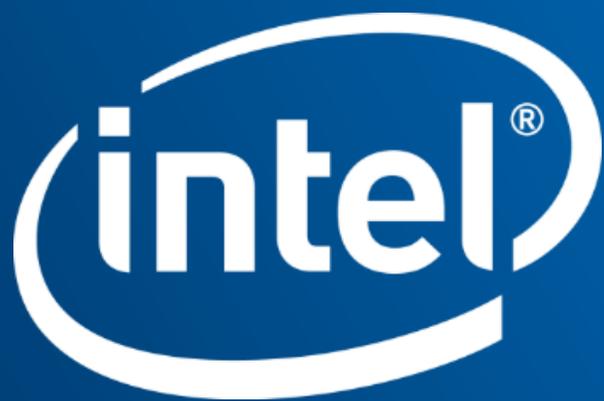
- Create one package that contains headers and libraries, for use by developers
- Create one package of SPDK applications (spdk\_tgt, perf, fio plugin)
- Unify the targets into a single application with a single management tool

**SPDK**

**SPDK-DEVEL**

# NEXT STEPS

- Work with major Linux distros to move forward with SPDK packages
  - RedHat/Centos/Fedora?
  - Debian/Ubuntu?
  - SuSE/SLES?



---

# BACKUP

---

