Current Status of OpenSSL

ICMC 2022
Wed 14-Sep-2022, 11:15

Ing. Martin Koci, MBA
OpenSSL Project Manager

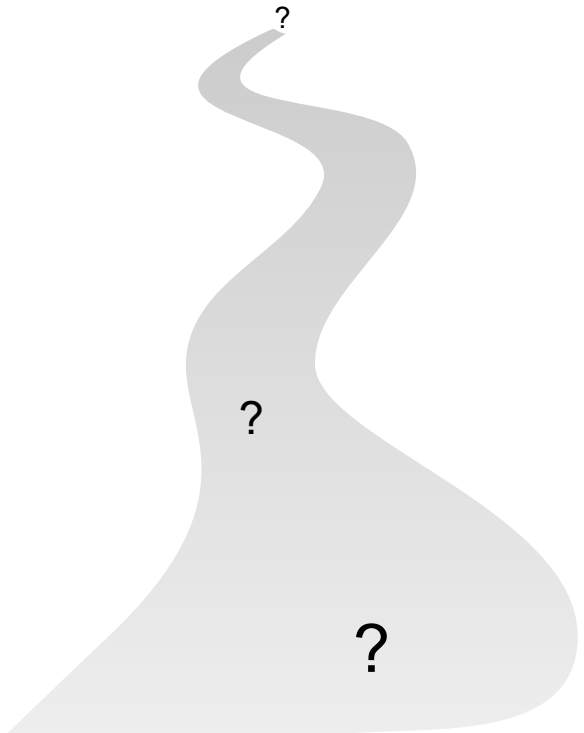
Agenda

- OpenSSL Vision Statement
- OpenSSL Roadmap
  - OpenSSL FIPS 140-2 validation certificate
  - OpenSSL FIPS 140-3 submission for validation
  - OpenSSL QUIC support
- Key Takeaways
- FAQ
- Resources and Links

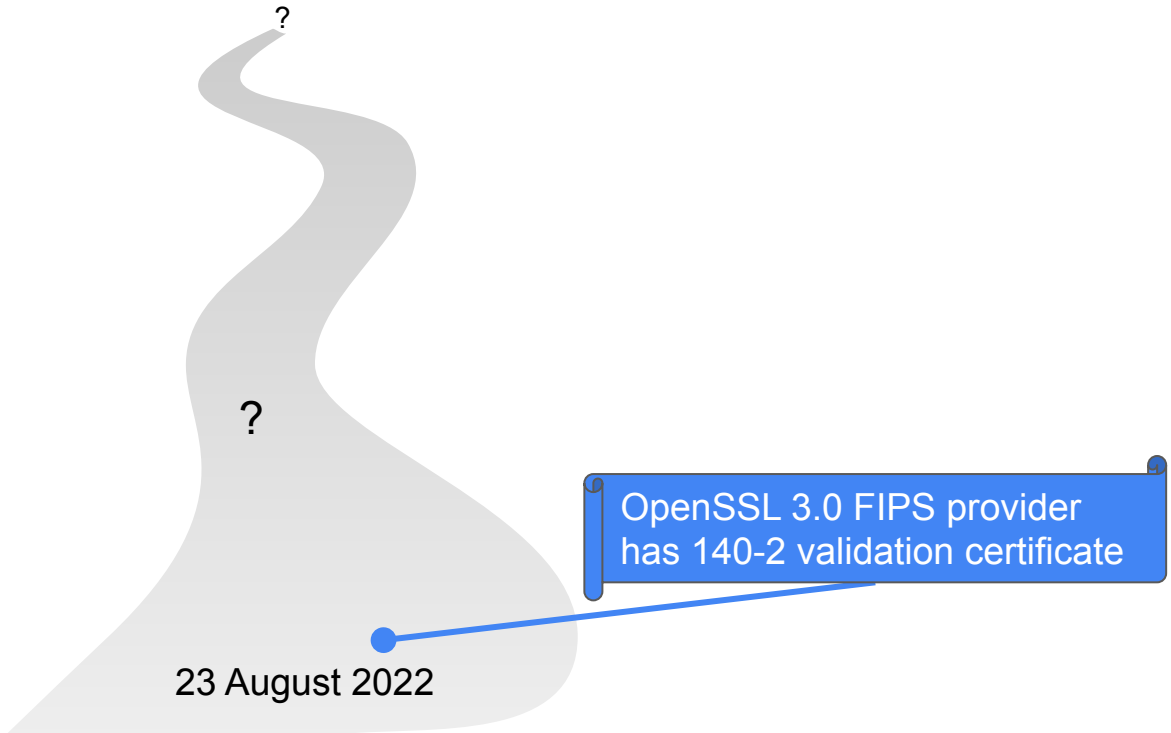
OpenSSL Vision Statement

"*Be a **trusted open-source software leader** in general-purpose cryptography and secure communication that meets **FIPS requirements for cryptographic modules for commercial needs**.*"

OpenSSL Roadmap



OpenSSL Roadmap



OpenSSL FIPS Provider 3.0

https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4282

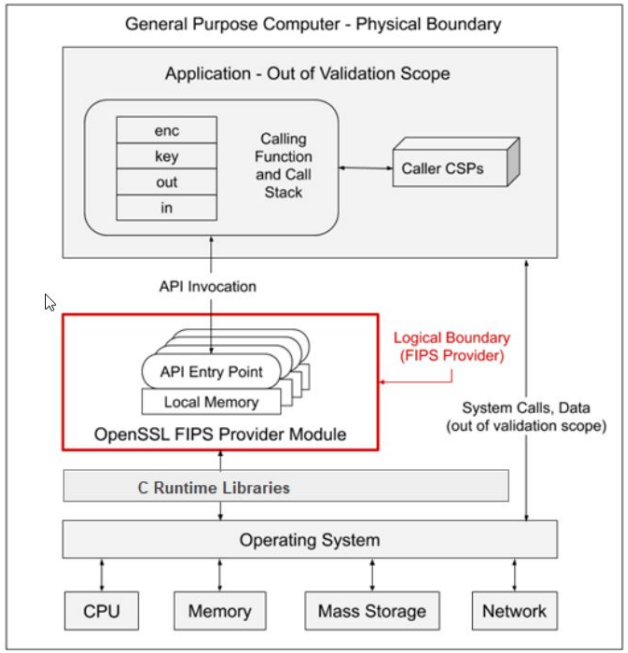


Figure 1 – Module Block Diagram

#	Operating System/Hypervisor	Hardware Platform	Processor	Optimizations (Target)
1	Photon OS 4.0 on ESXi 7.0	Dell PowerEdge R740	Intel Xeon Gold 6230R (x64)	PAA (AES-NI)
2	Windows Server 2019 on ESXi 7.0	Dell PowerEdge R740	Intel Xeon Gold 6230R (x64)	PAA (AES-NI)
3	Oracle Solaris 11.4	Oracle SPARC T8-1	Oracle SPARC M8-1 (x64)	PAA (SPARC)
4	Ubuntu Linux 20.04.1 Server	Dell Inspiron 7573	Intel i7 (x64)	PAA (AES-NI)
5	Windows 10	Dell Inspiron 7591	Intel i7 (x64)	PAA (AES-NI)
6	FreeBSD stable/13	NetApp HCI H700E	Intel Xeon E5-2695v4 (Broadwell) (x64)	PAA (AES-NI)
7	Debian 10	Fujitsu CX400 Blade	Intel Xeon Silver 4116 (Cascade Lake) (x64)	PAA (AES-NI)
8	Custom Ubuntu Linux 18.04	Akamai X8 system	Intel Xeon D1541 (x64)	PAA (AES-NI)
9	macOS 11.5.2	Apple M1 Mac Mini	M1	None
10	macOS 11.5.2	Apple M1 Mac Mini	M1	PAA (AES-NI)
11	macOS 11.5.2	Apple i5 Mac Mini	Intel i7	None
12	macOS 11.5.2	Apple i5 Mac Mini	Intel i7	PAA (AES-NI)

OpenSSL Old (Historical) FIPS Module

https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/1747

AES	Certs. #1884, #2116, #2234, #2342, #2394, #2484, #2824, #2929, #3090 and #3264
CVL	Certs. #10, #12, #24, #36, #49, #53, #71, #85, #260, #331, #372 and #472
DRBG	Certs. #157, #229, #264, #292, #316, #342, #485, #540, #607 and #723
DSA	Certs. #589, #661, #693, #734, #748, #764, #853, #870, #896 and #933
ECDSA	Certs. #264, #270, #315, #347, #378, #383, #394, #413, #496, #528, #558 and #620
HMAC	Certs. #1126, #1288, #1363, #1451, #1485, #1526, #1768, #1856, #1937 and #2063
RSA	Certs. #960, #1086, #1145, #1205, #1237, #1273, #1477, #1535, #1581 and #1664
SHS	Certs. #1655, #1840, #1923, #2019, #2056, #2102, #2368, #2465, #2553 and #2702
Triple-DES	Certs. #1223, #1346, #1398, #1465, #1492, #1522, #1695, #1742, #1780 and #1853

OpenSSL FIPS Provider 3.0

https://csrc.nist.gov/projects/Cryptographic-Algorithm-Validation-Program/details?source=A&number=1938

- AES-{CBC,ECB,OFB}
- AES-CBC-CS{1,2,3}
- AES-{CCM,CMAC,CTR}
- AES-CFB{1,8,128}
- AES-{GCM,GMAC,XTS}
- AES-{KW,KWP}
- Counter DRBG
- DSA KeyGen (FIPS186-4)
- DSA PQGGen (FIPS186-4)
- DSA PQGVer (FIPS186-4)
- DSA SigGen (FIPS186-4)
- DSA SigVer (FIPS186-4)
- ECDSA KeyGen (FIPS186-4)
- ECDSA KeyVer (FIPS186-4)
- ECDSA SigGen (FIPS186-4)
- ECDSA SigVer (FIPS186-4)
- Hash DRBG
- HMAC DRBG
- HMAC-SHA-1
- HMAC-SHA2-{224,256,384,512,**512/224,512/256**}
- **HMAC-SHA3-{224,256,384,512}**
- **KAS-ECC CDH SP800-56Ar3**
- **KAS-ECC-SSC SP800-56Ar3**
- **KAS-FFC-SSC SP800-56Ar3**
- **KAS-IFC-SCC**
- **KDA HKDF SP800-56Cr2**
- **KDA OneStep SP800-56Cr2**
- **KDA TwoStep SP800-56Cr2**
- **KDF ANS 9.42, 9.63**
- **KDF SP800-108**
- **PRF {SSH,TLS}**
- **KMAC-{128,256}**
- **KTS-IFC**
- **PBKDF**
- RSA KeyGen (FIPS186-4)
- RSA SigGen (FIPS186-4)
- RSA Signature Primitive
- RSA SigVer (FIPS186-4)
- Safe Primes {KeyGen,KeyVer}
- SHA-1
- SHA-2{224,256,384,512,**512/224,512/256**}
- **SHA-3{224,256,384,512}**
- **SHAKE-{128,256}**
- TDES-{CBC,ECB}
- **TLS v1.3 KDF**

- CKG (vendor affirmed)

Significant Architectural Changes

- Providers - contain implementation cryptographic algorithms
  - Default
  - Base
  - FIPS
  - Legacy
  - Third party
- Library contexts - where providers are loaded
  - No cross contamination between separate library contexts
  - Each is a stand alone “instance” of OpenSSL
  - A default library context is available for backwards compatibility
- Properties - how algorithm implementations are selected
  - Provide a mechanism to select implementations in a fine-grained manner
  - fips=”yes” guarantees that only FIPS algorithms will be used

Third party providers

- OQS [10] - Post Quantum Cryptography
  - Implements all of NIST's current PQC candidates
  - Partially integrated into TLS & the rest is coming
  - Available now
- GOST - Russian cryptography
  - In progress – ciphers and digests currently implemented
- TPM 2.0
  - Existing algorithms implemented outside of OpenSSL
  - Available now
- Blake 3
  - Available now

- Once standardization is complete, OQS and Blake 3 can/will be included in OpenSSL

Use of FIPS provider [1]

1. Via configuration
    a. FIPS only
    b. With extra algorithms
2. Via library calls (code)
    a. FIPS only
    b. With extra algorithms

Use of FIPS provider - via configuration (FIPS only)

The minimal configuration needed to enable FIPS mode:

```
openssl_conf = openssl_init

.include fipsmodule.cnf

[openssl_init]
providers = provider_sect

[provider_sect]
fips = fips_sect
base = base_sect

[base_sect]
activate = 1
```

Use of FIPS provider - via configuration (Extra algorithms)

```
openssl_conf = openssl_init

.include fipsmodule.cnf

[openssl_init]
providers = provider_sect
alg_section = evp_properties

[evp_properties]
default_properties = "fips=yes"

[provider_sect]
fips = fips_sect
default = default_sect

[default_sect]
activate = 1
```

Use of FIPS provider - via code (FIPS only)

```
OSSL_PROVIDER *fips;
OSSL_PROVIDER *base;
EVP_MD *dfips;

fips = OSSL_PROVIDER_load(NULL, "fips");
base = OSSL_PROVIDER_load(NULL, "base");


dfips = EVP_MD_fetch(NULL, “sha2-256”, NULL);

EVP_MD_free(dfips);

OSSL_PROVIDER_unload(base);
OSSL_PROVIDER_unload(fips);
```

Load FIPS and base providers into the default library context.

Load SHA2-256 implementation

Clean up at the end.

Use of FIPS provider - via code (Extra algorithms)

```
OSSL_PROVIDER *fips;
OSSL_PROVIDER *dflt;
EVP_MD *dfips, *dnonfips;

fips = OSSL_PROVIDER_load(NULL, "fips");
dflt = OSSL_PROVIDER_load(NULL, "default");
EVP_set_default_properties(NULL, "fips=yes");
```

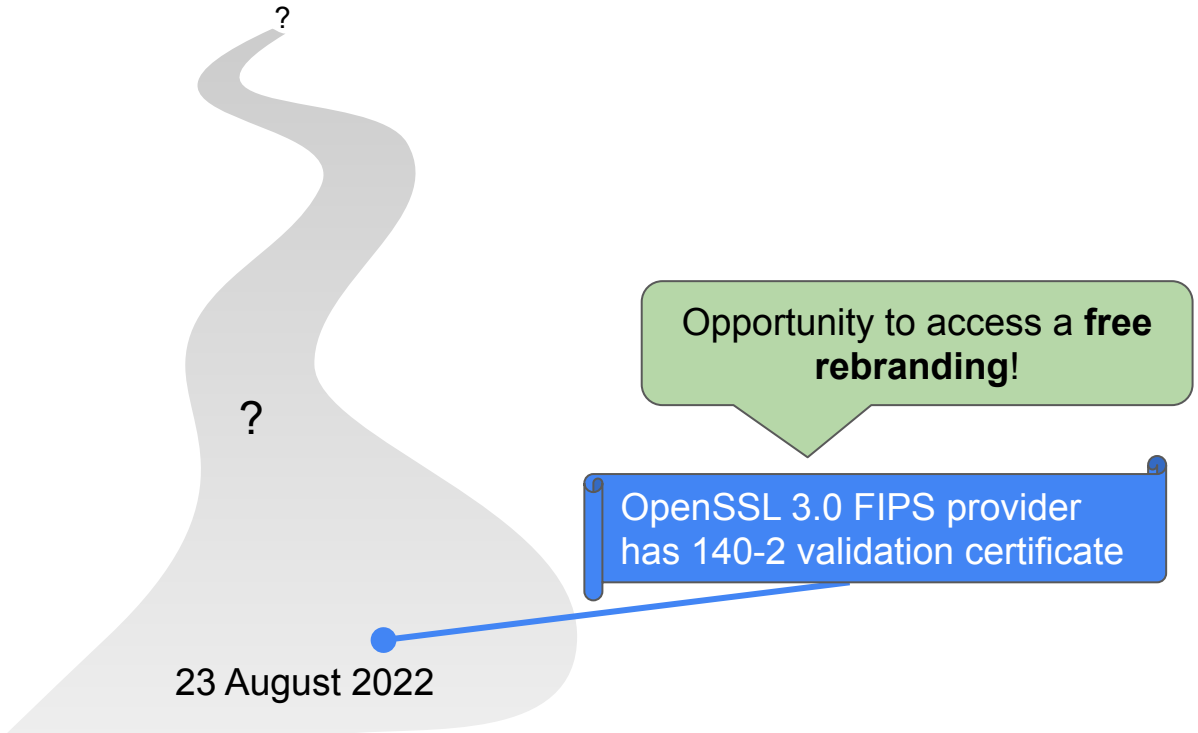
Load FIPS and default providers into the default library context.
Set default property query to ensure FIPS

```
dfips = EVP_MD_fetch(NULL, “sha2-256”, NULL);
dnonfips = EVP_MD_fetch(NULL, “md5”, “-fips”);

EVP_MD_free(dfips);  EVP_MD_free(dnonfips);

OSSL_PROVIDER_unload(dflt);
OSSL_PROVIDER_unload(fips);
```

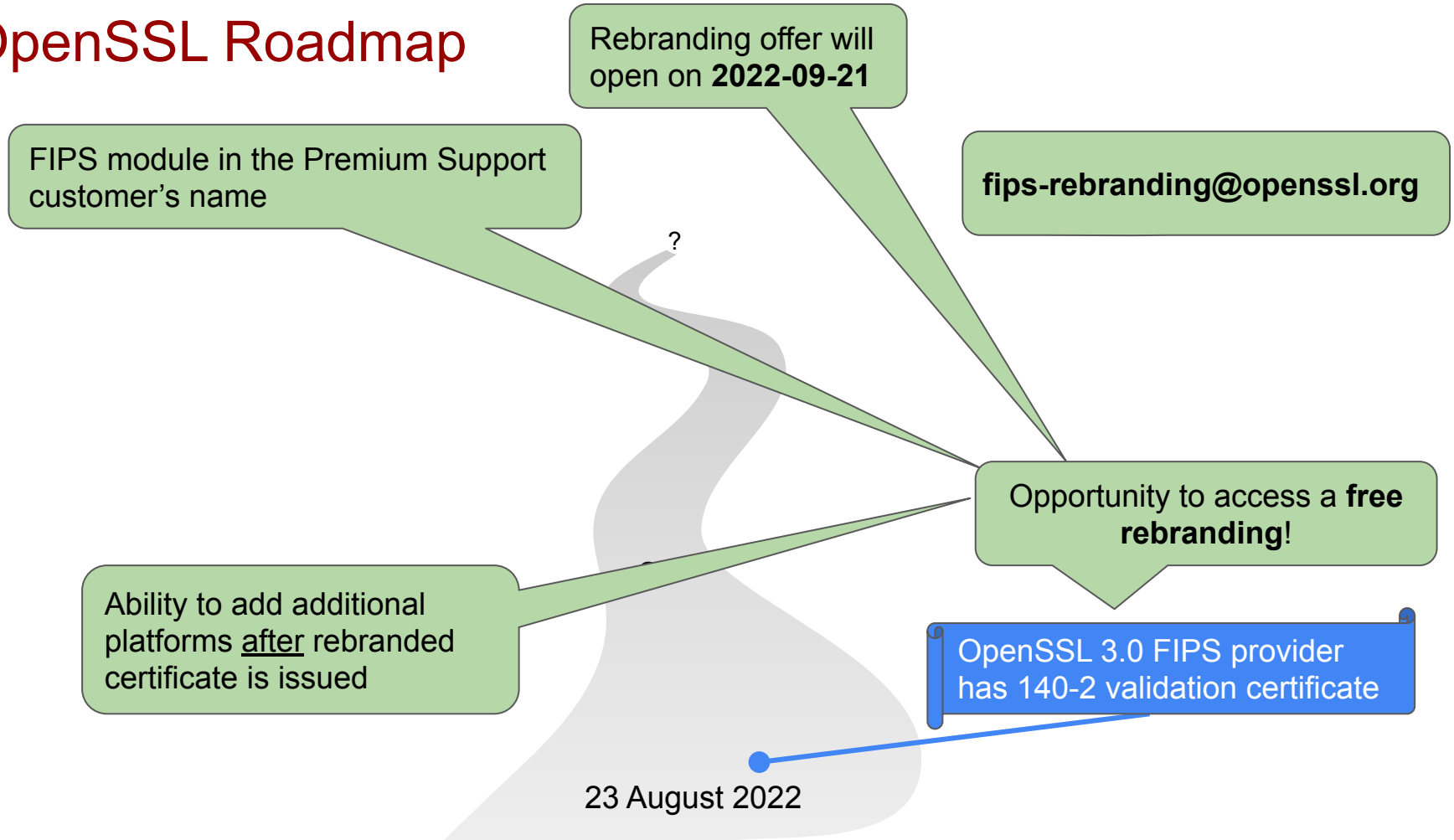
Load two digests, one FIPS and one not.

Clean up at the end.

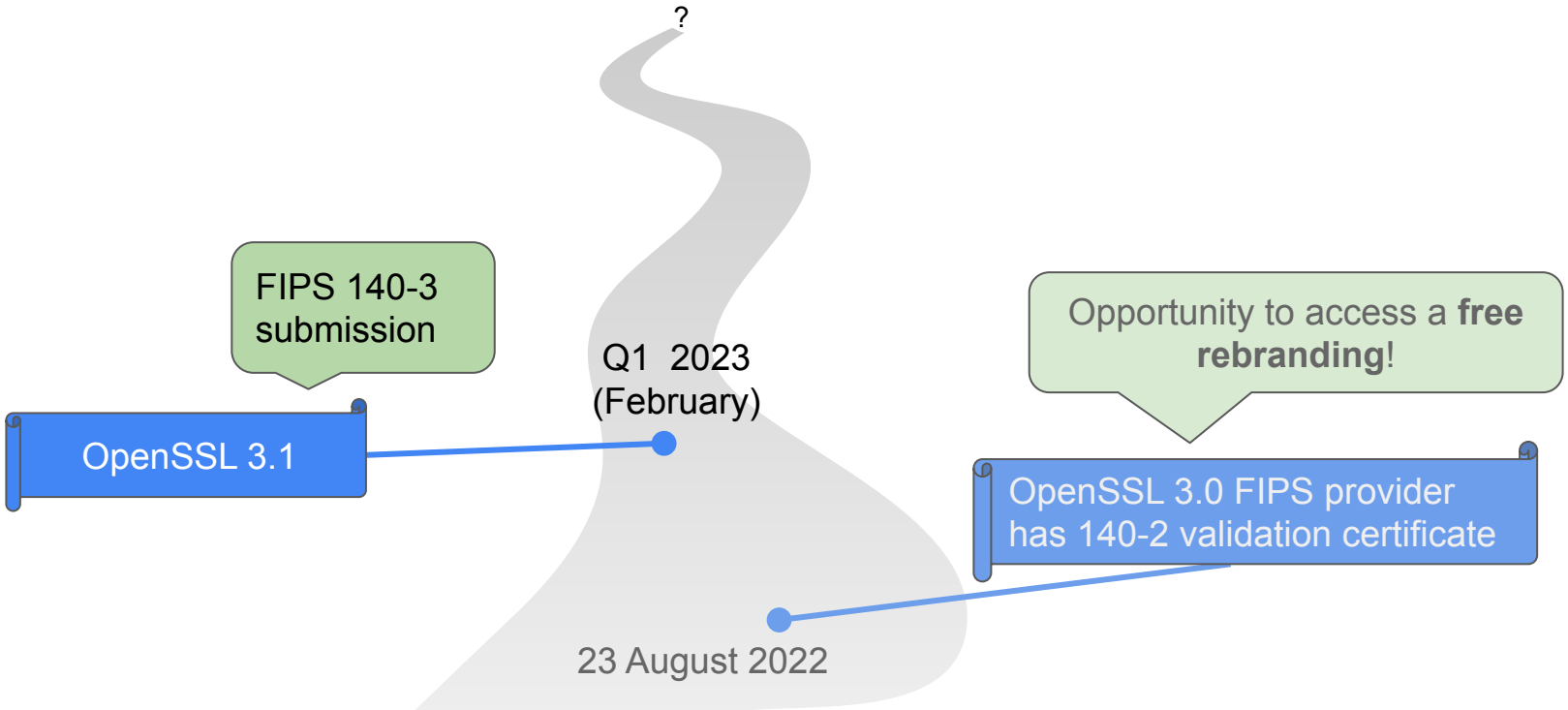
OpenSSL Roadmap



OpenSSL Roadmap



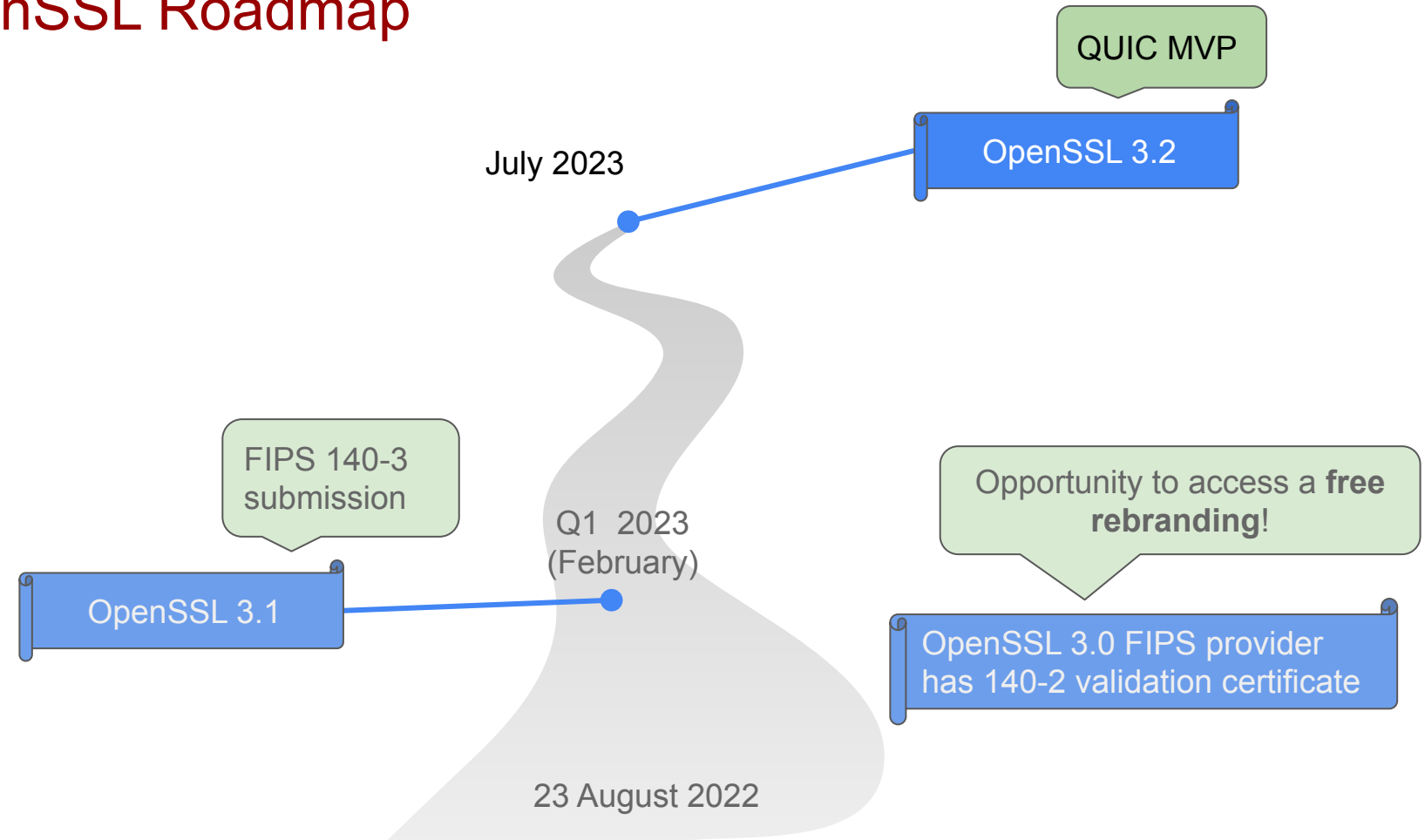
OpenSSL Roadmap



Change of Announced Roadmap Plan

- OpenSSL 3.1 was going to be the QUIC initial support release
  - We are pushing that to OpenSSL 3.2 (renaming the release)
- OpenSSL 3.1 will be the FIPS 140-3 submission release based on the current OpenSSL 3.0

OpenSSL Roadmap



Key takeaways

- ✓ Opportunity to access a free FIPS 140-2 **rebranding**!
- ✓ There is significant FIPS 140-2 items increase in the current module
- ✓ 140-2 FIPS provider can be used across minor versions due to change of architecture
- ✓ **The OpenSSL 3.1** release will be about **FIPS 140-3** validation submission
- ✓ Anybody else can implement a new crypto algorithm - 3th party provider
- ✓ **QUIC MVP** release pushed to **OpenSSL 3.2 release**

FAQ 1/2

1. What is needed to use OpenSSL 3.0 FIPS provider?
   - Information about how to configure and use the FIPS provider in your applications is available on the FIPS module man page [1]. You must also read the module security policy [2] and precisely follow the specific build and installation instructions included in it.
2. Can I do rebrand?
   - If you take out a premium support contract with us, you may get a single zero cost rebrand via Acumen Security [9] for a module in your own name (i.e. whitelabeled) on the condition that you agree to not further whitelabel it for other companies.
3. Is it possible to update the tested operational environments?
   - The OpenSSL project does not plan on adding any operational environments to our validation. You can do it separately by engaging with Acumen Security [9] (after you receive your rebranded module).
4. FIPS validation follow-up
   - We do not build a FIPS validation suite for distribution with OpenSSL. The right point of contact for follow-up validation work is Acumen Security [9].

FAQ 2/2

5. If operating environments are not listed, does that mean the certificate is not valid for these platforms? Is it possible to use OpenSSL 3.0 FIPS provider on our own operating platforms?
    - The certificate lists certain platforms as tested but it does not mean that you cannot use it on other platforms. You may simply be able to user affirm the module for another platform.
    - Certain agencies in Federal Government may have additional rules requiring the operational platform to be listed on the validation certificate.
6. Would we still have to work with Acumen Security [9] if our architecture matched the tested architecture?
    - No. If you are happy with the operational environments as listed on the certificate then nothing further needs to be done. Acumen Security [9] will still be involved because they are the lab we use for the rebranding itself.
7. Ongoing maintenance of the 3.0 FIPS provider?
    - The project has an agreement with Acumen Security [9] to review changes to the FIPS boundary for follow up maintenance work as required.

Resources and links

1. https://www.openssl.org/docs/man3.0/man7/fips_module.html
2. https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp4282.pdf
3. www.openssl.org
4. https://csrc.nist.gov/projects/Cryptographic-Algorithm-Validation-Program/details?source=A&number=1938
5. https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/1747
6. https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4282
7. fips-rebranding@openssl.org
8. https://keypair.us/
9. https://www.acumensecurity.net/
10. https://github.com/open-quantum-safe/oqs-provider