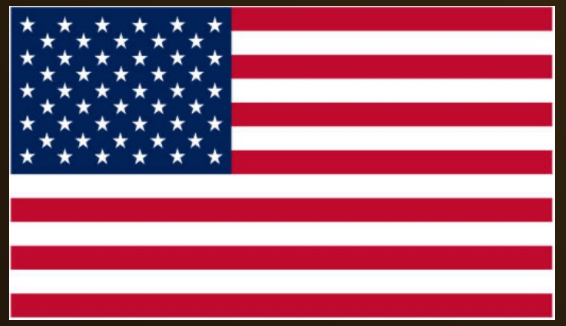
OpenSSL CONFERENCE

ONE MODULE THREE TRANSITIONS

§ PRIVACY AS A FUNDAMENTAL HUMAN RIGHT

20.10.2026



U.S.A.

20.10.2026

NSA and NIST

OpenSSL CONFERENCE

- FIPS 140-2 → FIPS 140-3
- CNSA1.0 → CNSA2.0
- NIST 2030 & 2035 Algorithm Transitions

Transitions are happening in all Jurisdictions

20.10.2026



European Union

OpenSSL CONFERENCE



- Germany (BSI)
  - TR-02102 Technical Guideline Framework updated 2026
  - Cloud Computing Compliance Criteria Catalogue, C5:2026
  - AIS 20/31 transition



- France (ANSSI)
  - ANSSI-PG-083 v3.00
  - ANSSI-CC-NOTE-24 and AIS20/31



- Spain
  - CCN-STIC 221
  - CCN-STIC 807

Without a Cryptographic Center of Excellence

20.10.2026



OpenSSL CONFERENCE

With A Cryptographic Center Of Excellence

20.10.2026



How Can Organizations Build Applications Without Replicating Cryptographic Policy Within Each Application Seperately?

OpenSSL Policy Levers

- Build Configuration
- OpenSSL Global Configuration
- Provider Configuration
- Query Properties
- TLS/SSL
  - Security Level
  - CipherSuite Selection
  - Key Exchange Groups
- Contexts
  - Library
  - SSL
  - Signing
  - Key Gen
- Authentication



Build Policies

FIPS Policy

- Build the fips module.
- Does NOT mean you are now FIPS compliant.
- You must be using a CMVP validated FIPS provider
- -pedantic for highest level of conformance during fipsinstall

Algorithm Policy

- no-des
- no-rc2, no-rc4, no-rc5
- no-md2, no-md4
- no-dsa
- no-legacy provider

Protocol Policy

- no-weak-ssl-ciphers
- no-tls1 no-tls1_1
- no-tls1-method no-tls1_1-metho
- no-dtls1 no-dtls1-method

API Policy

OpenSSL Compilation

- no-deprecated
- enabled-deprecated
- --api=x.y[.z]

Application Compilcation

-DOPENSSL_API_COMPAT

-DOPENSSL_NO_DEPRECATED

OpenSSL CONFERENCE

Deploying Build Policy



20.10.2026

Setting FIPS Policy In Configuration

```
# policy-fips-routing.cnf
config_diagnostics = 1
openssl_conf = openssl_init
.include /absolute/path/to/generated/fipsmodule.cnf

[openssl_init]
providers = provider_sect
alg_section = algorithm_sect

[provider_sect]
fips = fips_sect
base = base_sect

[base_sect]
activate = 1

[algorithm_sect]
default_properties = fips=yes
```



Setting TLS Policy In Configuration

```
# policy-128-bit-security.cnf
openssl_conf = openssl_init

[openssl_init]
providers = provider_sect
ssl_conf = ssl_sect

[ssl_sect]
system_default = tls_policy

[tls_policy]
# TLS 1.2 and 1.3 only.
MinProtocol = TLSv1.2
MaxProtocol = TLSv1.3
Groups = X25519MLKEM768:P-256:P-384
SignatureAlgorithms = ecdsa_secp256r1_sha256:ecdsa_secp384r1_sha384:rsa_pss_rsae_sha256:rsa_pss_rsa
Options = -Compression,NoRenegotiation,-UnsafeLegacyRenegotiation,-UnsafeLegacyServerConnect,-KTLS
# Minimum 128-bit security.
CipherString = DEFAULT:@SECLEVEL=3
# TLS 1.3 cipher-suite allowlist.
Ciphersuites = TLS_AES_256_GCM_SHA384:TLS_AES_128_GCM_SHA256
# Require an ephemeral key exchange during session resumption.
Options = -AllowNoDHEKEX
```



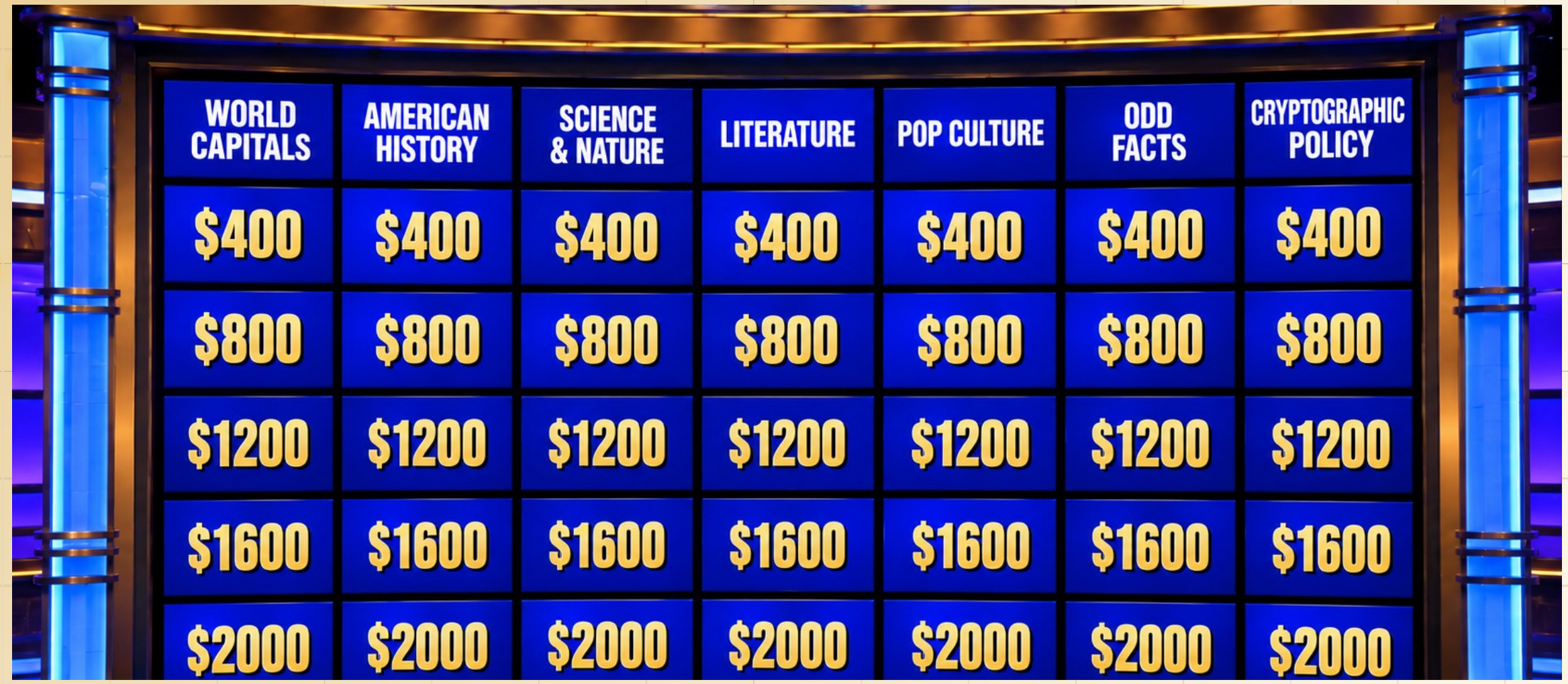
OpenSSL
CONFERENCE

Configuration Deployment Strategies



Jeopardy Round 1

OpenSSL CONFERENCE



OpenSSL CONFERENCE

Does Deploying Config Give You Control Over the App?

20.10.2026

```
int main(void)
    sha1 = NULL;

    default_provider = OSSL_PROVIDER_load(NULL, name: "default");

[openssl_init]
providers = provider_sect
alg_section = evp_properties

[evp_properties]
# Ensure FIPS non-approved algorithms in the FIPS module are
# TEST-RAND). This also means that EVP_default_properties_is
# returns the expected value
default_properties = "fips=yes"

[provider_sect]
fips = fips_sect

    sha1 = EVP_MD_fetch(NULL, algorithm: "SHA1", properties: "provider=default");
    if (!hash_and_print(sha1, expected_provider: "default", default_message))
        goto cleanup;
```

It does NOT!

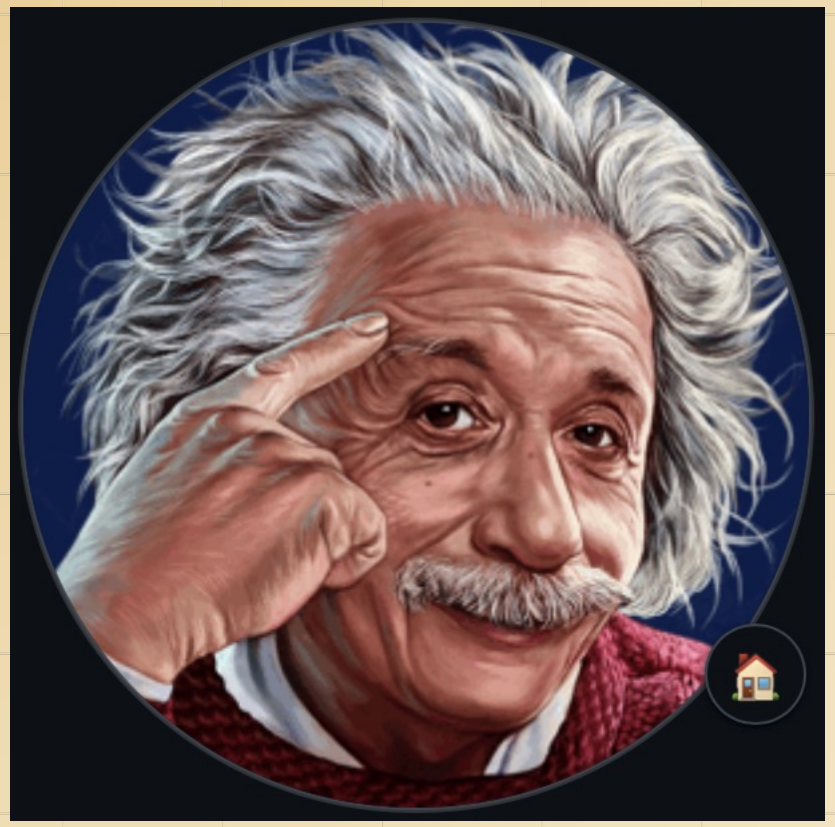
OpenSSL CONFERENCE

```
/Users/jake/workspace/safelogic/sl-140-3-openssl-3.5/cmake-build-debug/config_control
Default properties: fips=yes
SHA1 provider: fips
Message: I'm getting hashed by the FIPS provider
SHA1: 01f95f9221d3cc00dcc9d6ddfe27bf3340eeaa65
Default properties: -fips
SHA1 provider: default
Message: I'm not getting hashed by the FIPS provider
SHA1: 4fc8a56af7baad7156ef9d5c6d4a76650535cf4e

Process finished with exit code 0
```

Thought Experiment

OpenSSL CONFERENCE



The Multi Product Developement House

AS A CRYPTOGRAPHIC CENTER OF EXCELLENCE I WANT TO:

- Promote Strong Cryptographic Posture in Systems and Applications
- Prevent developers from using cryptographic anti-patterns in code
- Prevent the use of insecure algorithms

TECHNIQUE:

- Hide Crypto Libraries along with their complexity
- Build **Common Set of Cryptographic APIs**
- Enhance Your SDK With Policy Handling Capabilities

Product A – Fully Featured Web Storefront

Two Deployment Scenarios

- Commercial
  - Supports highly diverse set of clients
- Government
  - Support only authorized clients with strict security policy.

Minimal CryptoKit Example

Public C SDK

Include `<cryptokit.h>` and link to `libcryptokit`. The header exposes opaque listener and connection handles; applications do not need OpenSSL headers. `CRYPTOKIT_API_VERSION` is 2.

CALL	PURPOSE
`ck_init_fips`	Initialize the default OpenSSL context for CNSA2 or PQC. Optional before the matching listener call, which initializes implicitly.
`ck_listen_commercial`	Bind a commercial TLS listener.
`ck_listen_cnsa2`	Bind the government/CNSA2 listener.
`ck_listen_pq_only`	Bind the PQC listener.
`ck_listener_accept`	Accept TCP, complete TLS, enforce the profile, and return a checked `ck_connection`. Returns `CK_AGAIN` after an idle one-second poll.
`ck_connection_read` `ck_connection_write`	Transfer application bytes. A successful call may transfer fewer bytes than requested.
`ck_connection_info`	Return negotiated protocol, suite, group, server signature, and client-authentication result for an accepted connection.
`ck_policy_info` `ck_fips_status`	Return the selected process policy and FIPS/default-provider state. `ck_fips_status` reports a policy error if the FIPS property is inactive.
`ck_connection_close`	Release an accepted connection.
`ck_listener_stop`	Stop accepting and release the listener; existing returned connections remain owned by the application.
`ck_shutdown`	Release no global policy state; the process-wide OpenSSL context stays selected until exit.
`ck_last_error`	Read a thread-local diagnostic after a failed SDK call.

Static OpenSSL with Position Independent Code

OpenSSL CONFERENCE

```
export OPENSSL_PRIVATE="$PWD/_ck_openssl"

./Configure linux-x86_64 enable-fips no-shared -fPIC \
    --prefix="$OPENSSL_PRIVATE" \
    --libdir=lib

make -j"$(nproc)"
make test
make install_sw install_fips
```

Define a Public CryptoKit API

OpenSSL CONFERENCE

```
#ifndef CRYPTOKIT_H
#define CRYPTOKIT_H

/* GCC/Clang visibility declaration for this Linux example. */
#define CK_API __attribute__((visibility("default")))

#ifdef __cplusplus
extern "C" {
#endif

CK_API ck_status ck_listen_cnsa2(const ck_server_options *, ck_listener **);
CK_API ck_status ck_listen_pq_only(const ck_server_options *, ck_listener **);
CK_API ck_status ck_listen_commercial(const ck_server_options *, ck_listener **);

/* Types and remaining CK_API declarations are in the full header. */

#ifdef __cplusplus
}
#endif

#endif /* CRYPTOKIT_H */
```

cryptokit/exports.linux

```
{
    global:
        ck_init_fips;
        ck_fips_status;
        ck_policy_info;
        ck_listen_commercial;
        ck_listen_cnsa2;
        ck_listen_pq_only;
        ck_listener_accept;
        ck_connection_read;
        ck_connection_write;
        ck_connection_info;
        ck_connection_close;
        ck_listener_stop;
        ck_shutdown;
        ck_last_error;

    local:
        *;
};
```

Building and Linking

OpenSSL CONFERENCE

```
gcc -O2 -fPIC -fvisibility=hidden \
    -Icryptokit/include \
    -I"$OPENSSL_PRIVATE/include" \
    -c cryptokit/src/cryptokit.c -o cryptokit.o

gcc -shared -o libcryptokit.so \
    cryptokit.o \
    "$OPENSSL_PRIVATE/lib/libssl.a" \
    "$OPENSSL_PRIVATE/lib/libcrypto.a" \
    -Wl --version-script=cryptokit/exports.linux \
    -Wl --exclude-libs,ALL \
    -Wl,-z,defs \
    -ldl -pthread
```

Verify Results

OpenSSL CONFERENCE

```
→  cryptokit-sdk-install nm -gU "$ck_sdk_prefix/lib/libcryptokit.dylib"
0000000000002a04 T _ck_connection_close
0000000000002be8 T _ck_connection_info
0000000000002a60 T _ck_connection_read
0000000000002b3c T _ck_connection_write
0000000000001848 T _ck_fips_status
0000000000000f80 T _ck_init_fips
0000000000002d28 T _ck_last_error
0000000000001dd8 T _ck_listen_cnsa2
00000000000018b0 T _ck_listen_commercial
0000000000001e08 T _ck_listen_pq_only
0000000000001e38 T _ck_listener_accept
0000000000002c4c T _ck_listener_stop
00000000000015dc T _ck_policy_info
0000000000002d24 T _ck_shutdown
```

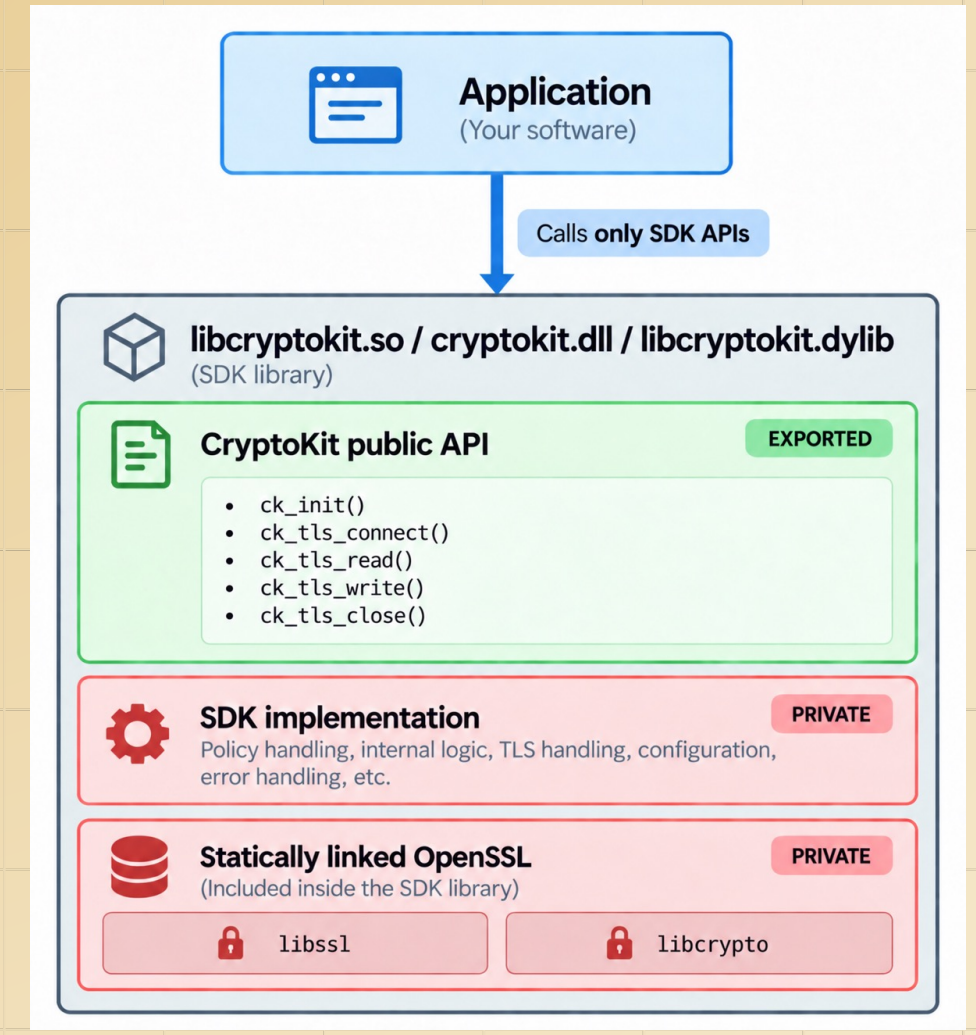
```
→  cryptokit-sdk-install otool -L lib/libcryptokit.dylib
lib/libcryptokit.dylib:
        @rpath/libcryptokit.dylib (compatibility version 0.0.0, current version 0.0.0)
        /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1356.0.0)
```

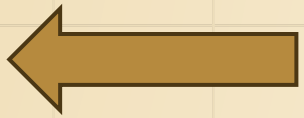
Application Build

OpenSSL CONFERENCE

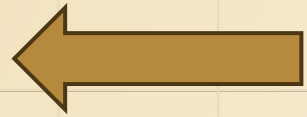
```
ck_sdk_prefix=/private/tmp/cryptokit-sdk-install
cc -std=c11 -I"$ck_sdk_prefix/include" cryptokit/demo/server.c \
  -L"$ck_sdk_prefix/lib" -Wl,-rpath,"$ck_sdk_prefix/lib" \
  -lcryptokit -o "$ck_sdk_prefix/bin/cryptokit_demo"
```

Separation of Concerns





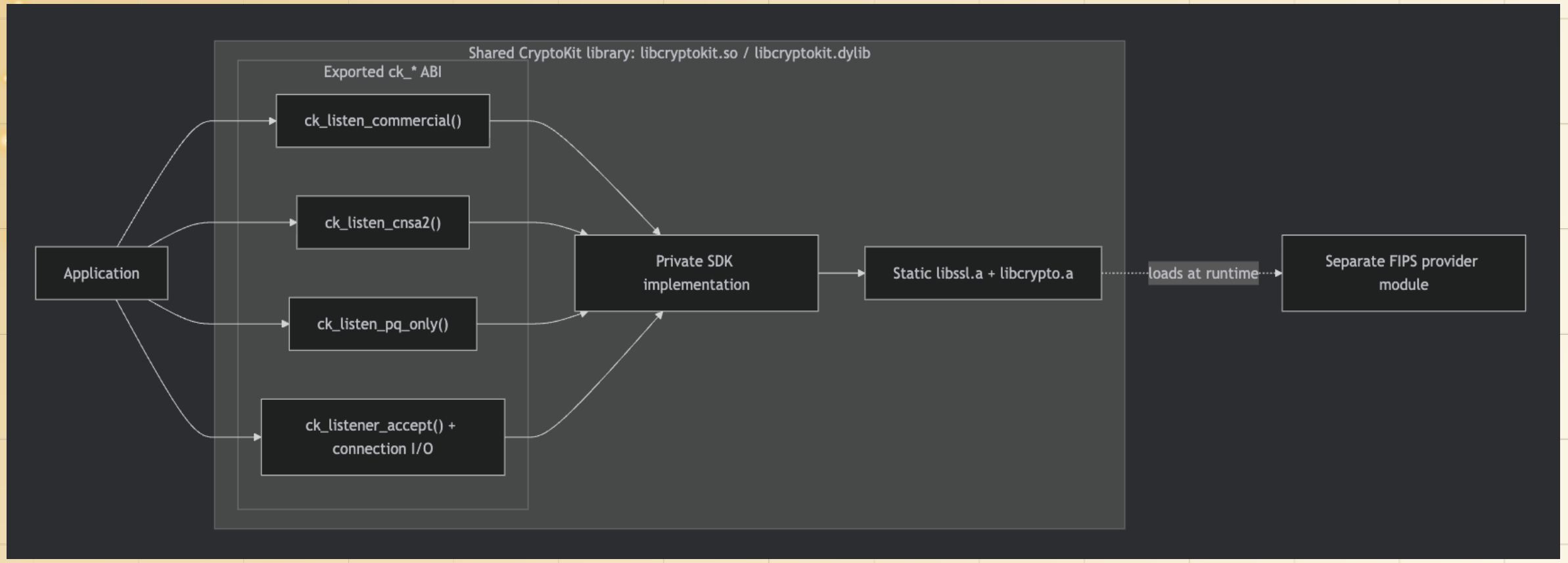






Application Integration

OpenSSL CONFERENCE



Application Policy Deployment Models

OpenSSL CONFERENCE

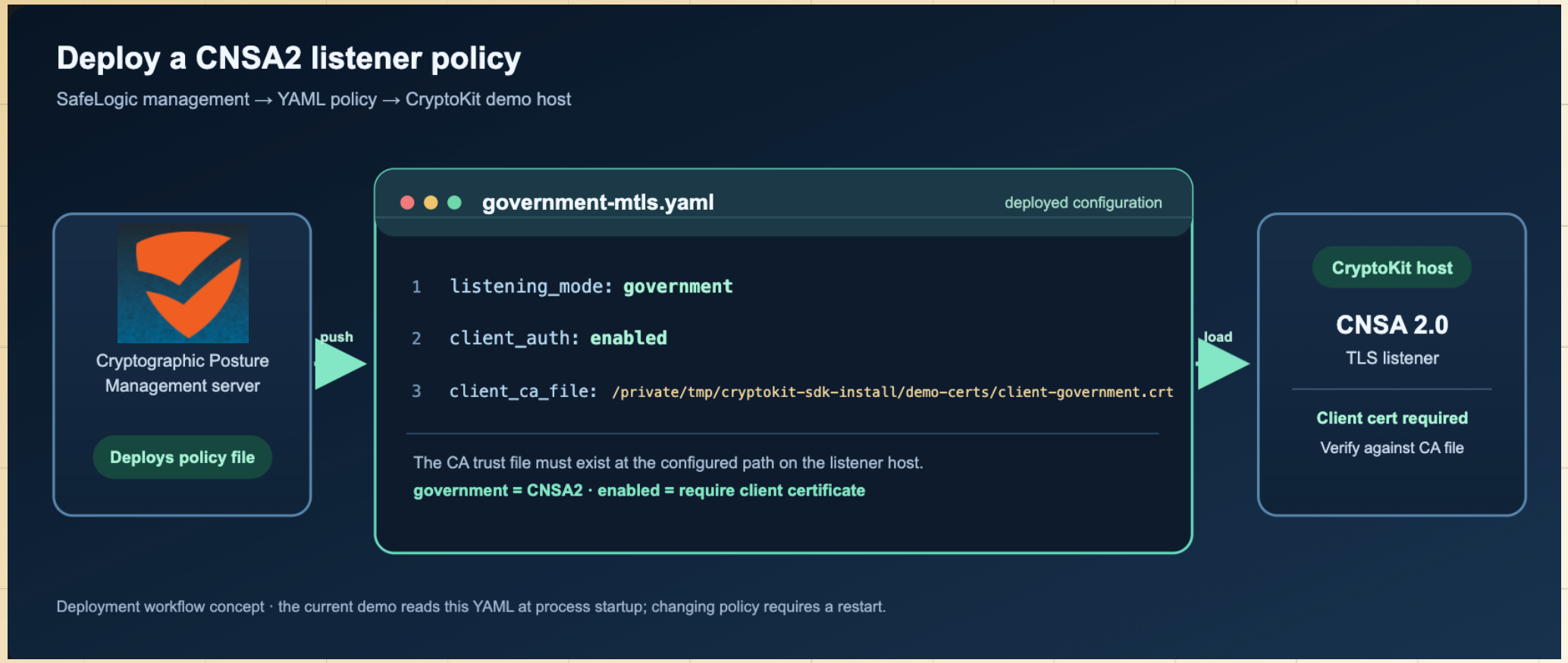
- **First Time Use Configured (On-Premise Installation)**
  - Admin at terminal selecting policy enforcement options

```
-- Install configuration: "Debug"
-- Installing: /private/tmp/cryptokit-sdk-install/lib/libcryptokit.dylib
-- Installing: /private/tmp/cryptokit-sdk-install/bin/cryptokit_demo
-- Up-to-date: /private/tmp/cryptokit-sdk-install/include/cryptokit.h
-- Up-to-date: /private/tmp/cryptokit-sdk-install/share/cryptokit/api.html
CryptoKit demo installed at /private/tmp/cryptokit-sdk-install
Choose listening mode (commercial/government/pqc): government
Require client authentication for CNSA2? [y/N]: y
Trusted client CA file [/private/tmp/cryptokit-sdk-install/demo-certs/client-government.crt]:
CryptoKit government demo listening on https://127.0.0.1:8443/
```

Application Policy Deployment Models

OpenSSL CONFERENCE

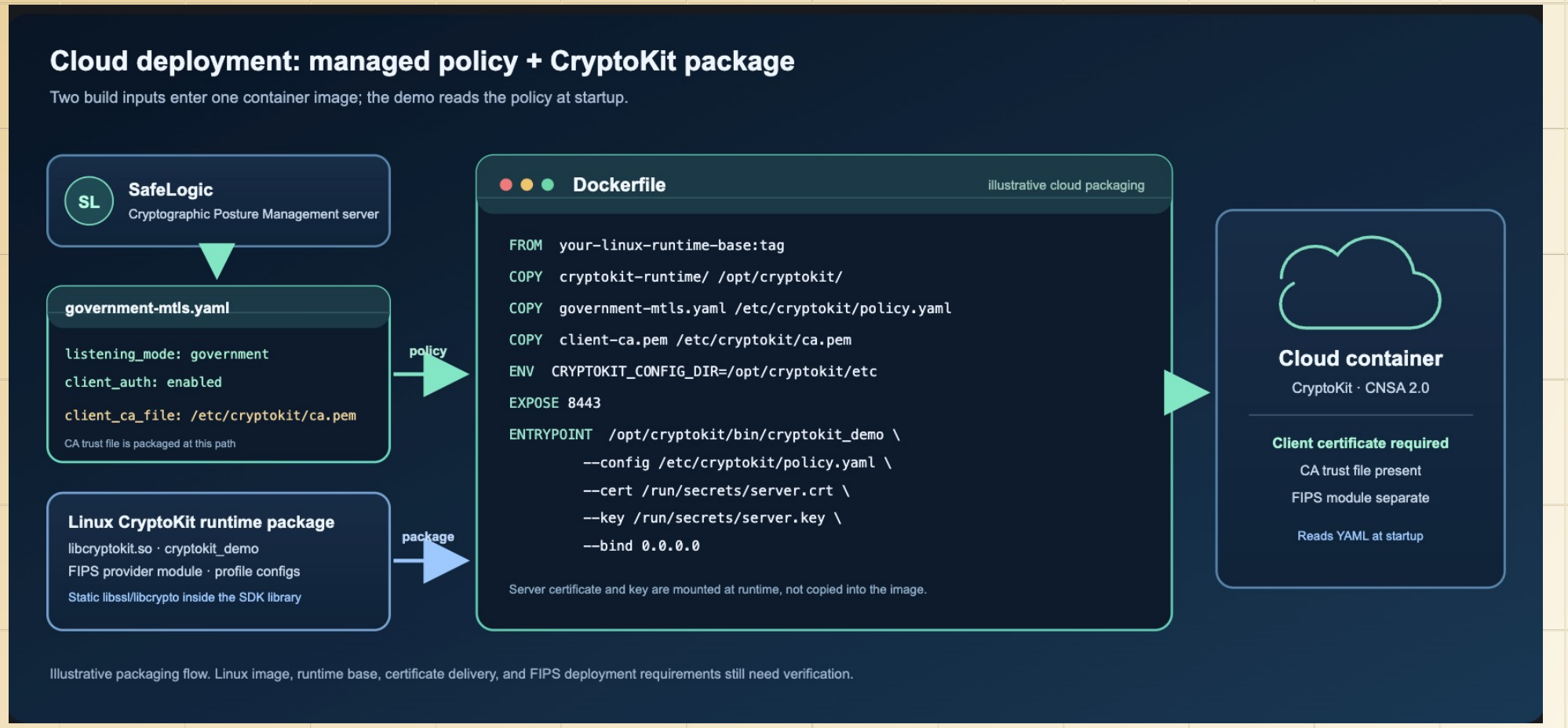
- **Device/Posture Management (Enterprise Managed IT)**
  - Deploy Policy via Centralized Management Service During Device Enrollment



Application Policy Deployment Models

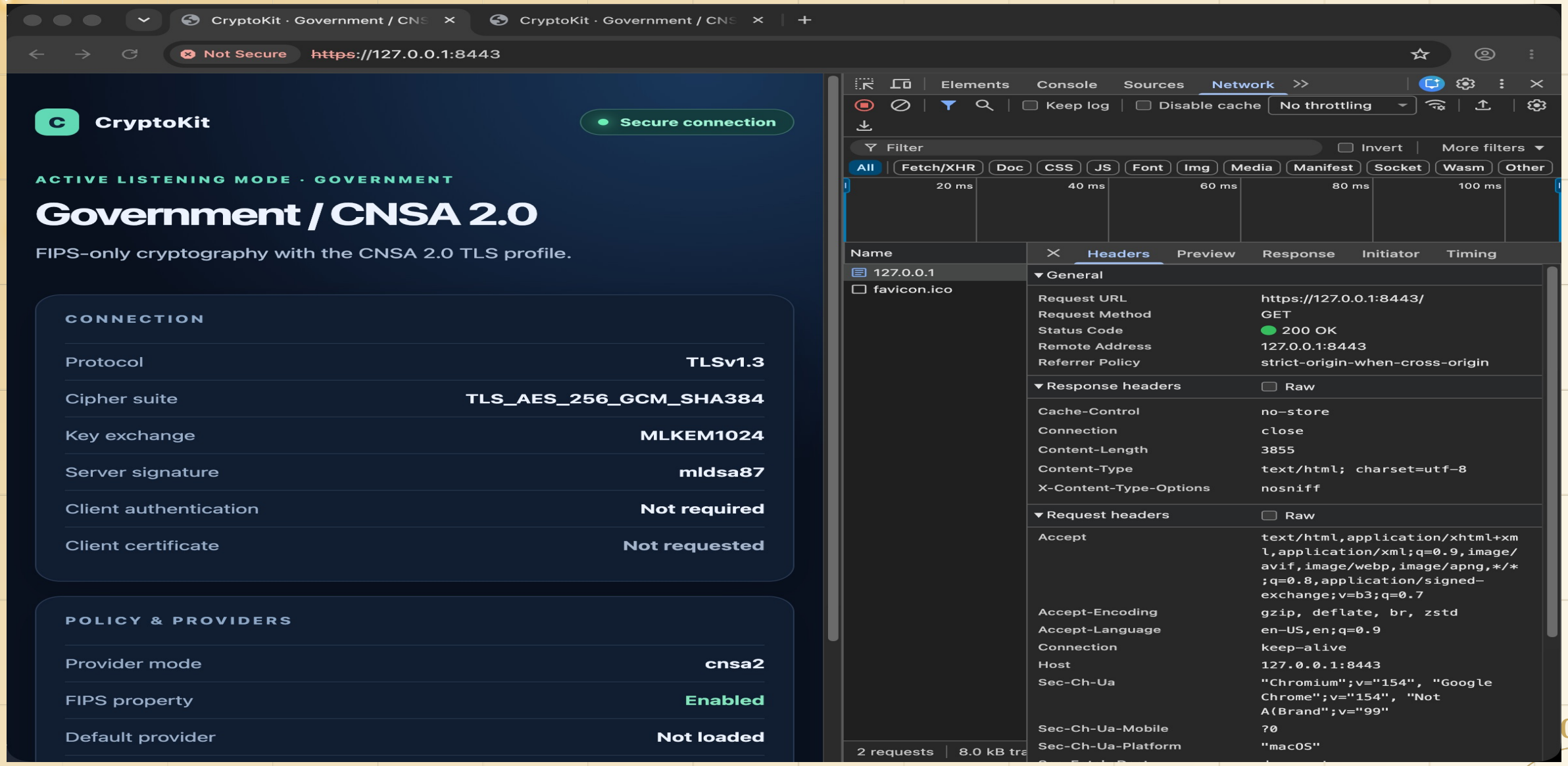
OpenSSL CONFERENCE

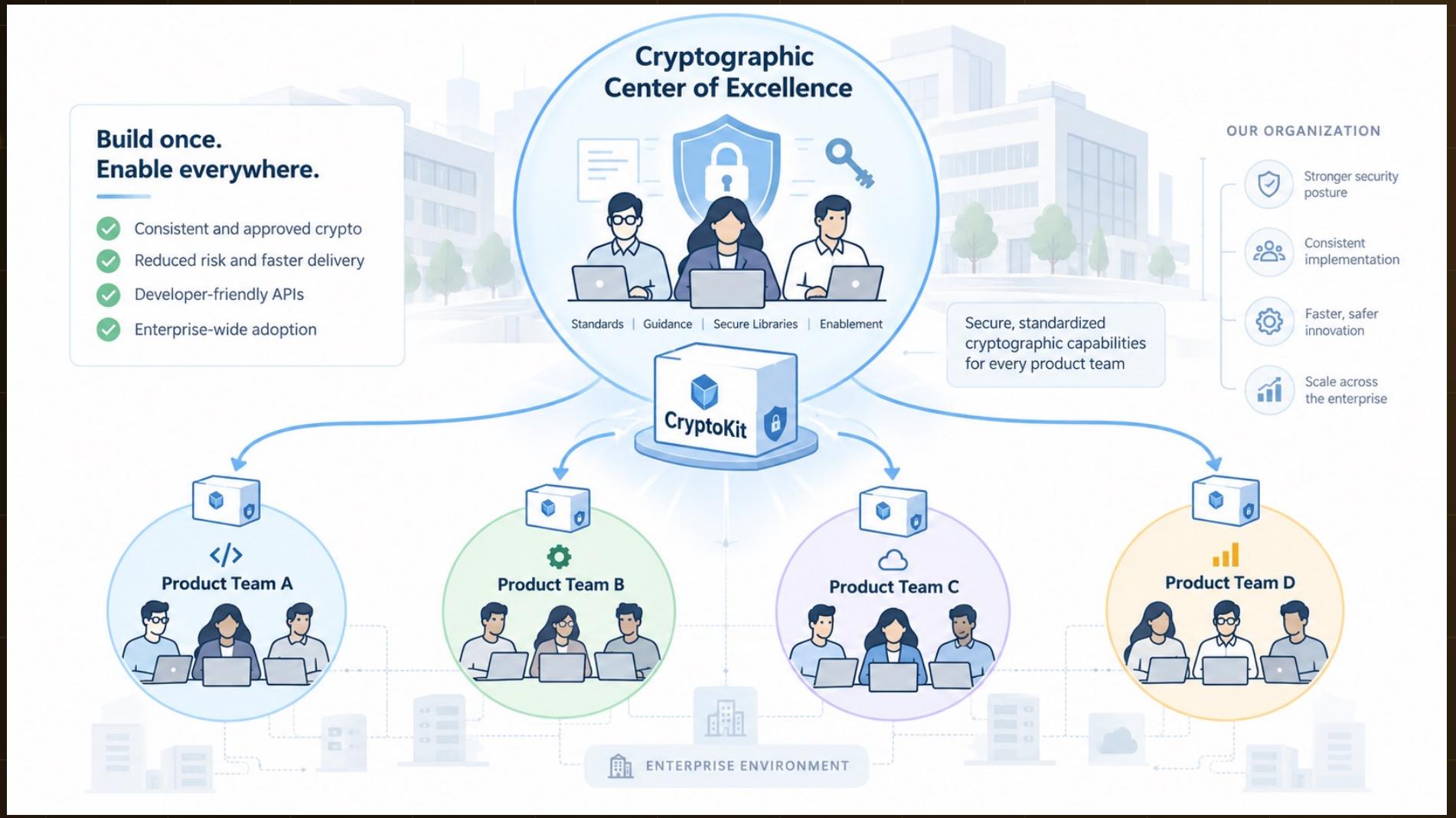
- **CICD Deployment (DevSecOps)**
  - Yaml file distributed with application replaces FTU selections.



Example

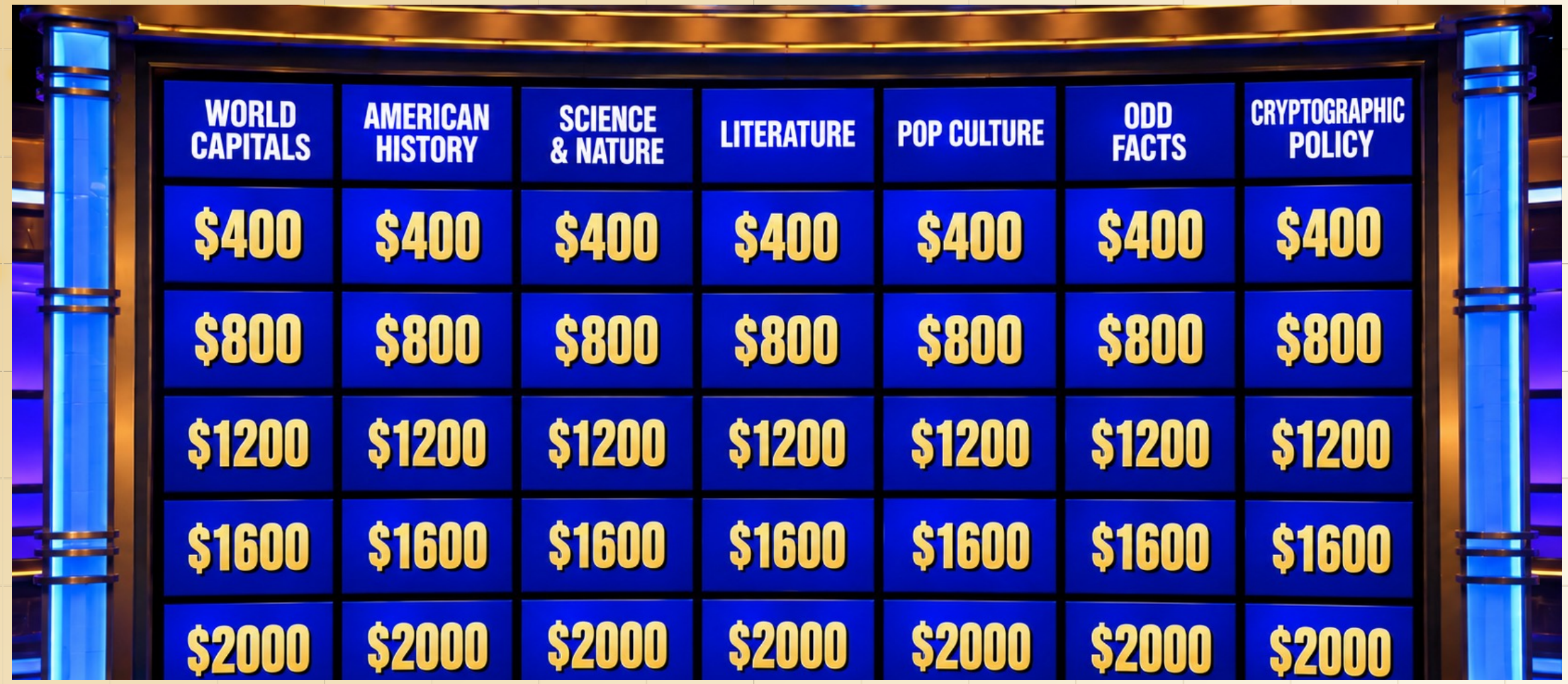
OpenSSL CONFERENCE




Cryptographic Center of Excellence
Standards | Guidance | Secure Libraries | Enablement
Build once.
Enable everywhere.
Consistent and approved crypto
Reduced risk and faster delivery
Developer-friendly APIs
Enterprise-wide adoption
Secure, standardized cryptographic capabilities for every product team
OUR ORGANIZATION
Stronger security posture
Consistent implementation
Faster, safer innovation
Scale across the enterprise
CryptoKit
Product Team A
Product Team B
Product Team C
Product Team D
ENTERPRISE ENVIRONMENT

Final Jeopardy

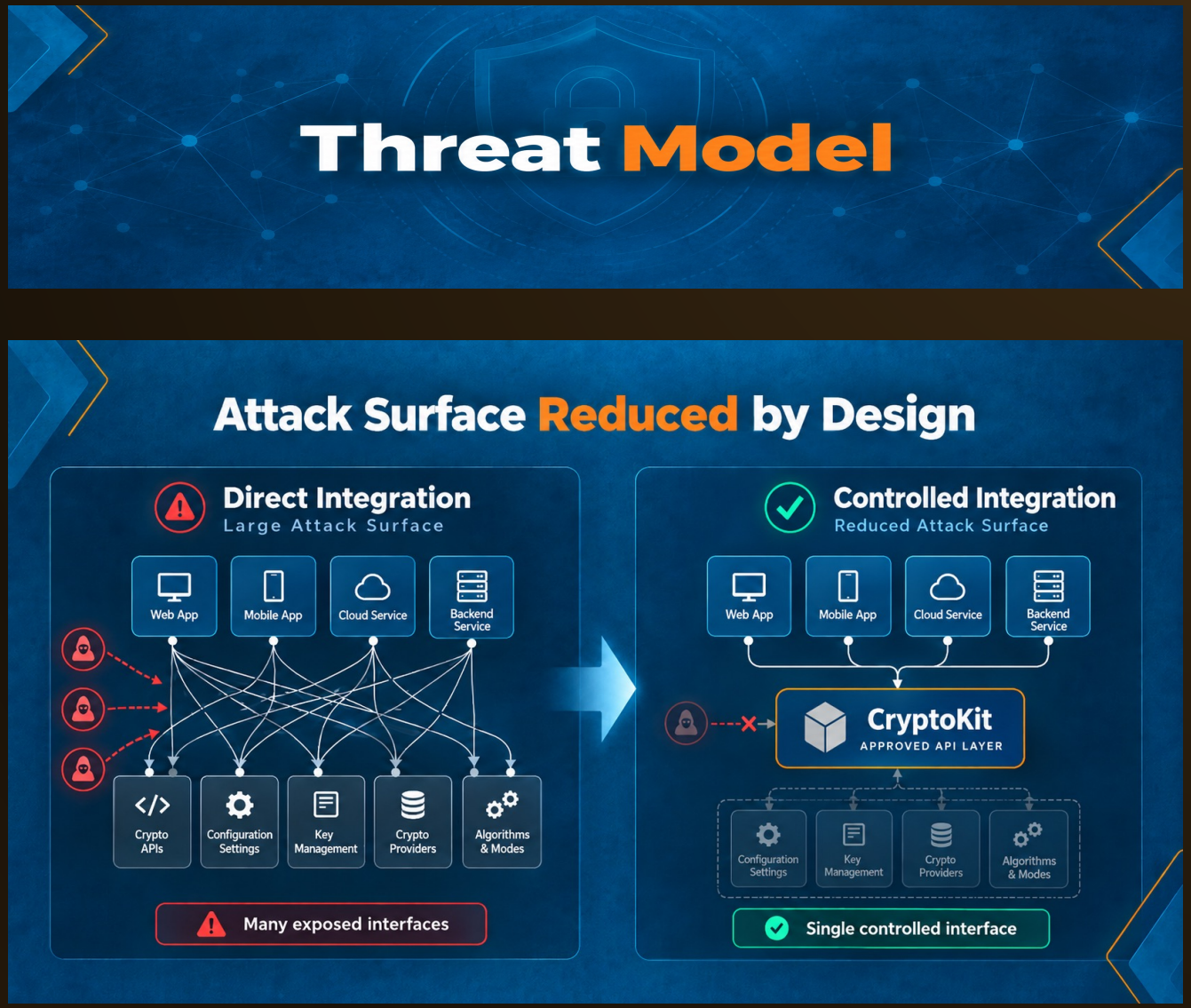
OpenSSL CONFERENCE



OpenSSL CONFERENCE

Have We Increased or Decreased our attack surface at the crypto layer?

20.10.2026





SafeLogic

Cryptography Simplified