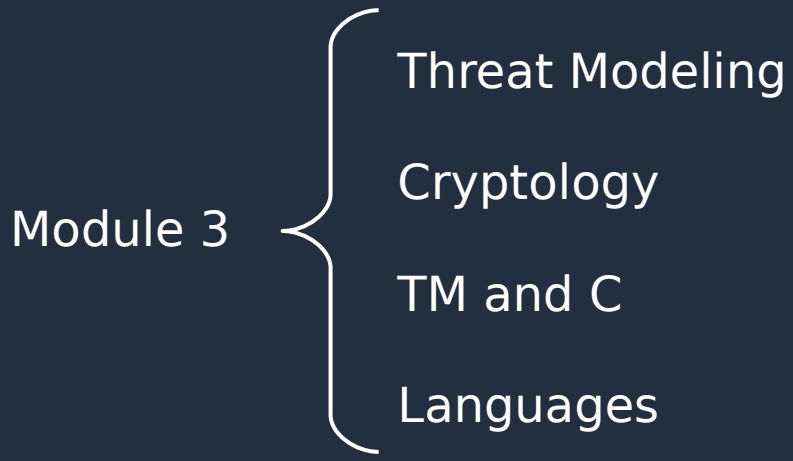
Welcome

- About this adventure:
  github.com/CuboCyberSecurity

- Consider all these modules as an introduction and as a way of sharing some key ideas (some *puzzle pieces*).

- It is a *free version* training:
  - Incomplete and imprecise information.
  - If you get confused, do not run away and make your own interpretations.

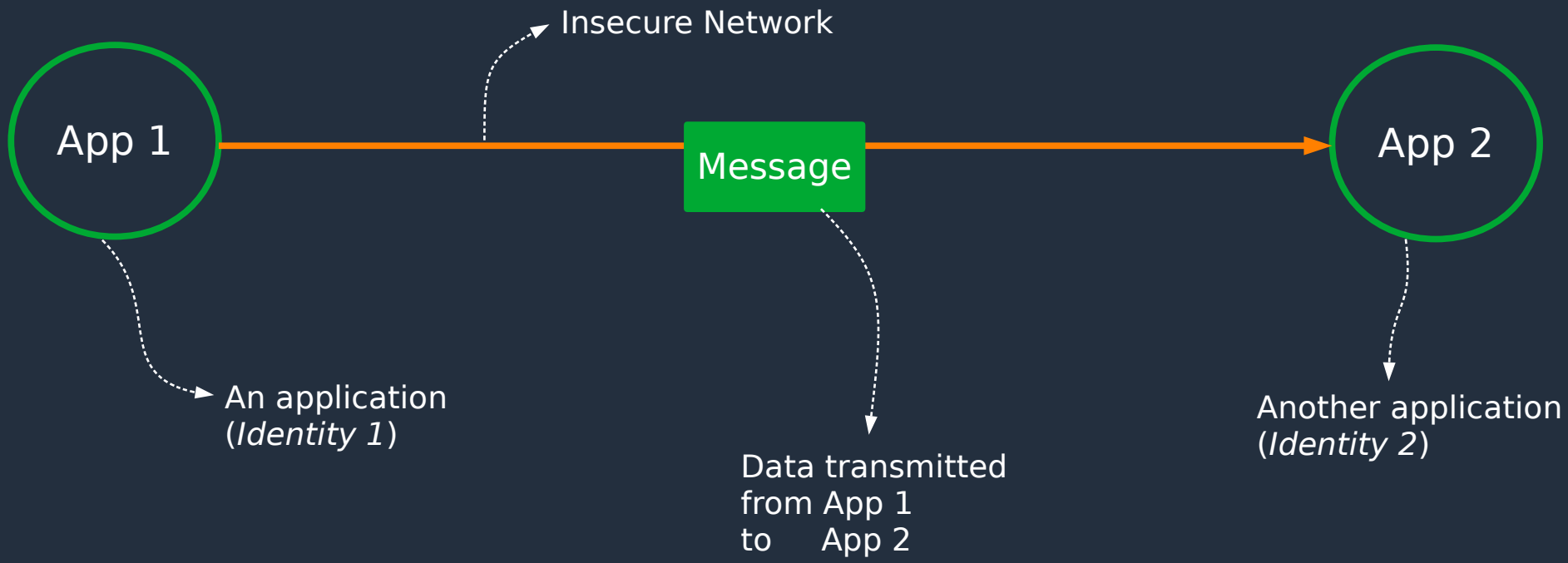
- Eventually, I'm going to share a much more complete and clear picture on the Wiki.

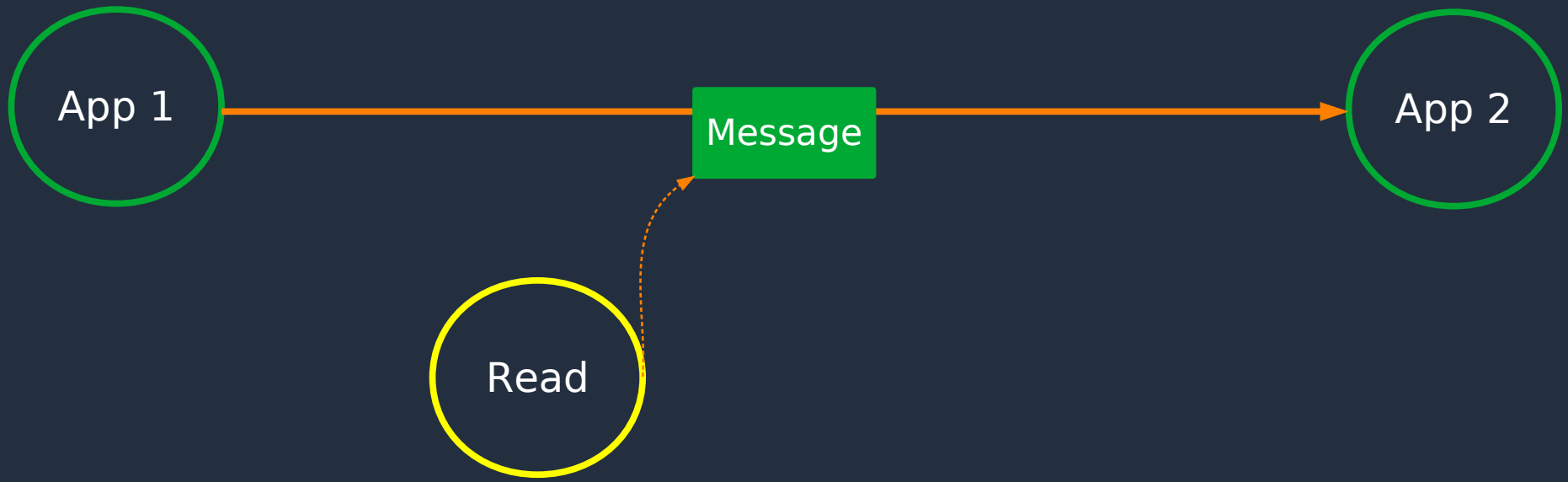

Module 3
Threat Modeling
Cryptology
TM and C
Languages

Threat Modeling

Tools we are going to use for this example:

- Security Principles and Security Policy.
- Data Lifecycle (CRUD).
- Access Control.

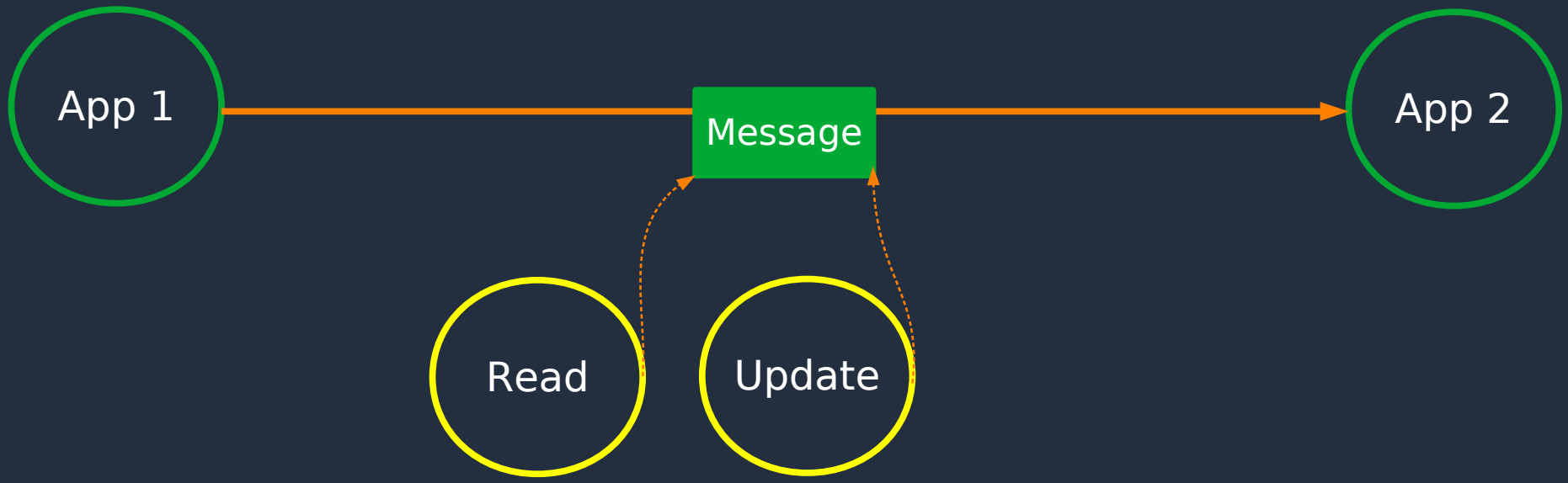

Insecure Network
App 1
Message
App 2
An application
(*Identity 1*)
Data transmitted
from App 1
to App 2
Another application
(*Identity 2*)



Threat
Read.

Reason
Confidentiality violation.

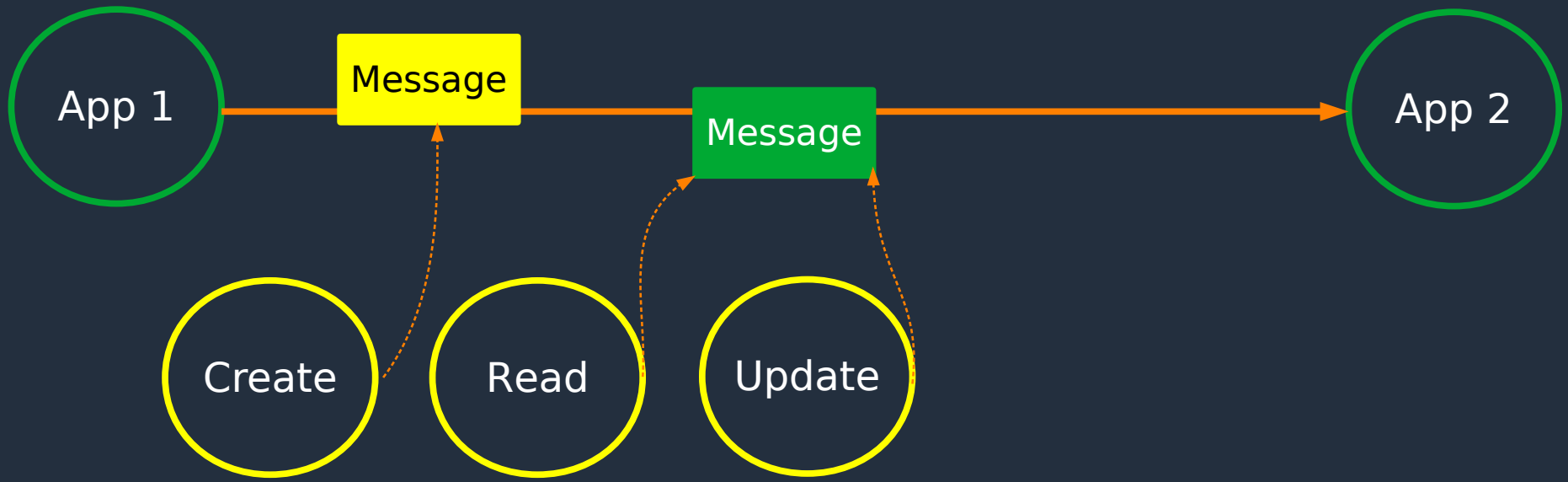
Mitigation
-



Threat
Update.

Reason
Integrity violation.

Mitigation
-



Threat

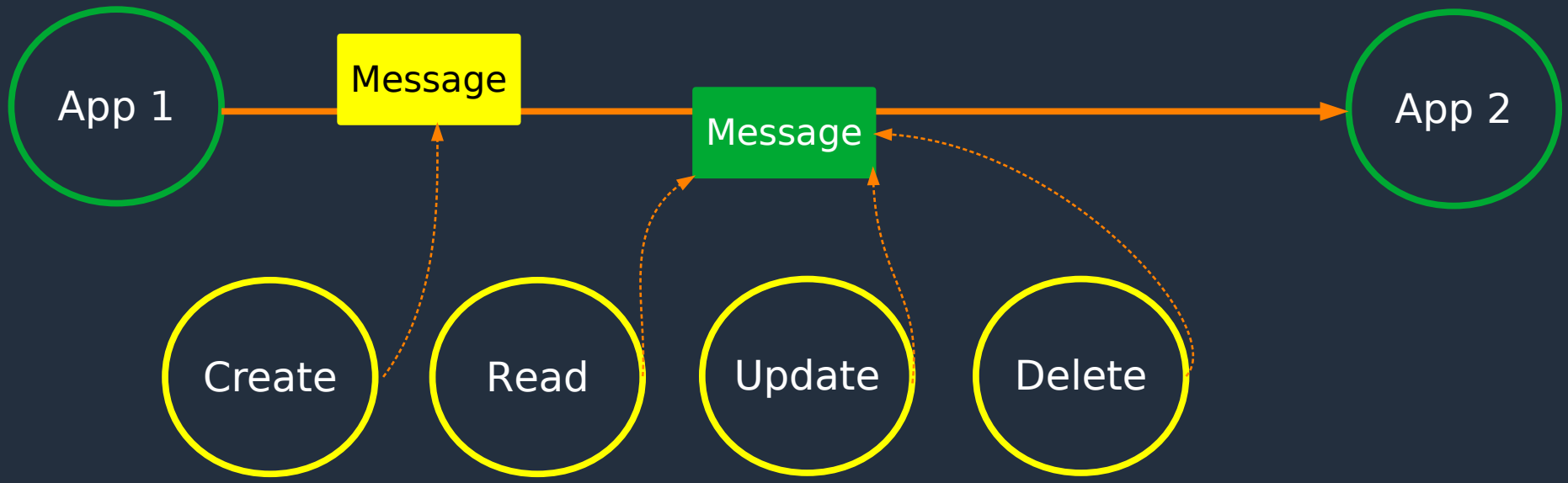
Create.

Reason

Authentication violation.

Mitigation

-



Threat

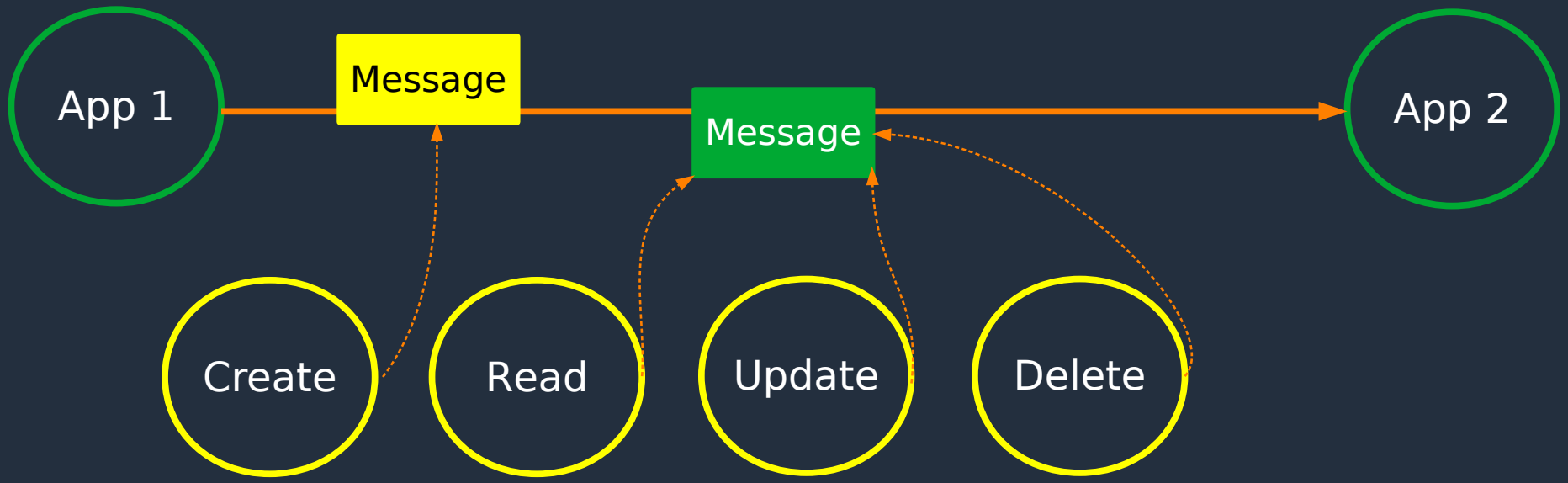
Delete.

Reason

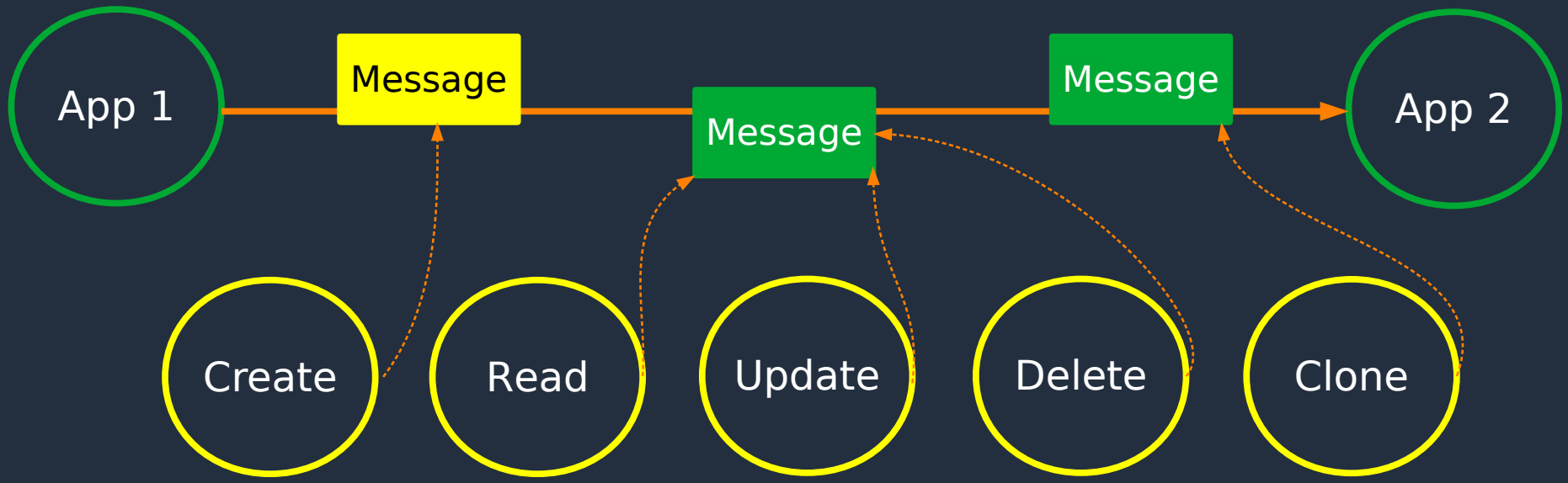
- Integrity violation.
- Availability violation.

Mitigation

-



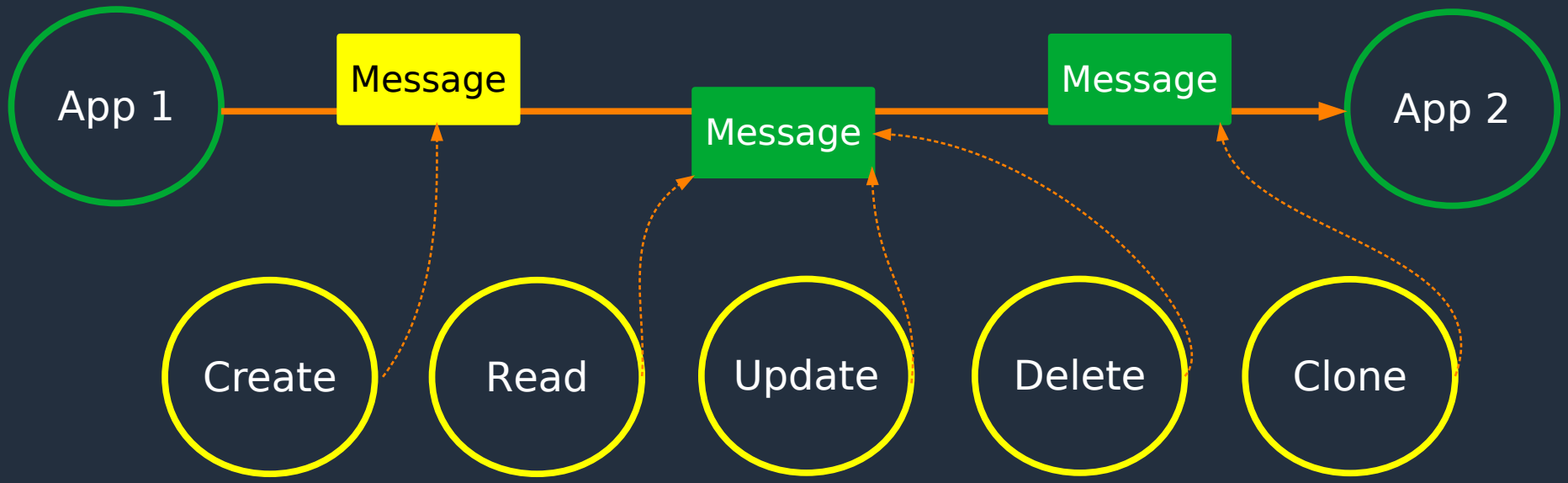
Done? ... No way!



Threat
Clone.

Reason
Non-Repudiation violation.

Mitigation
-



Have we identified **all** possible threats?

What if we add a third app ...

Threat Modeling Methods

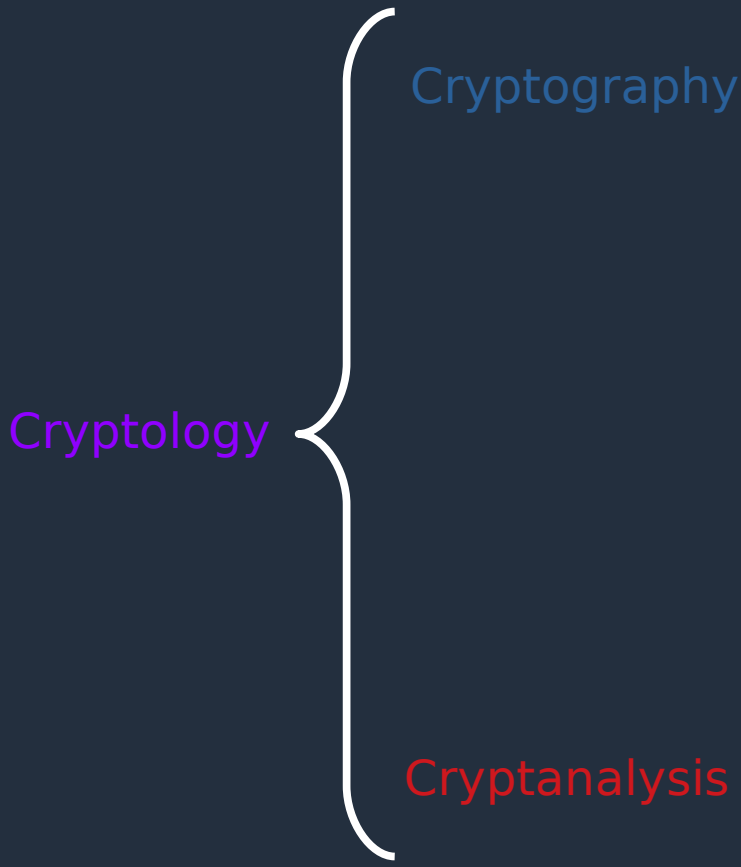
- STRIDE.
- PASTA.
- Trike.
- Attack Trees.
- CRUD.
- Security Cards.
- OCTAVE.
- VAST Modeling.
- ...

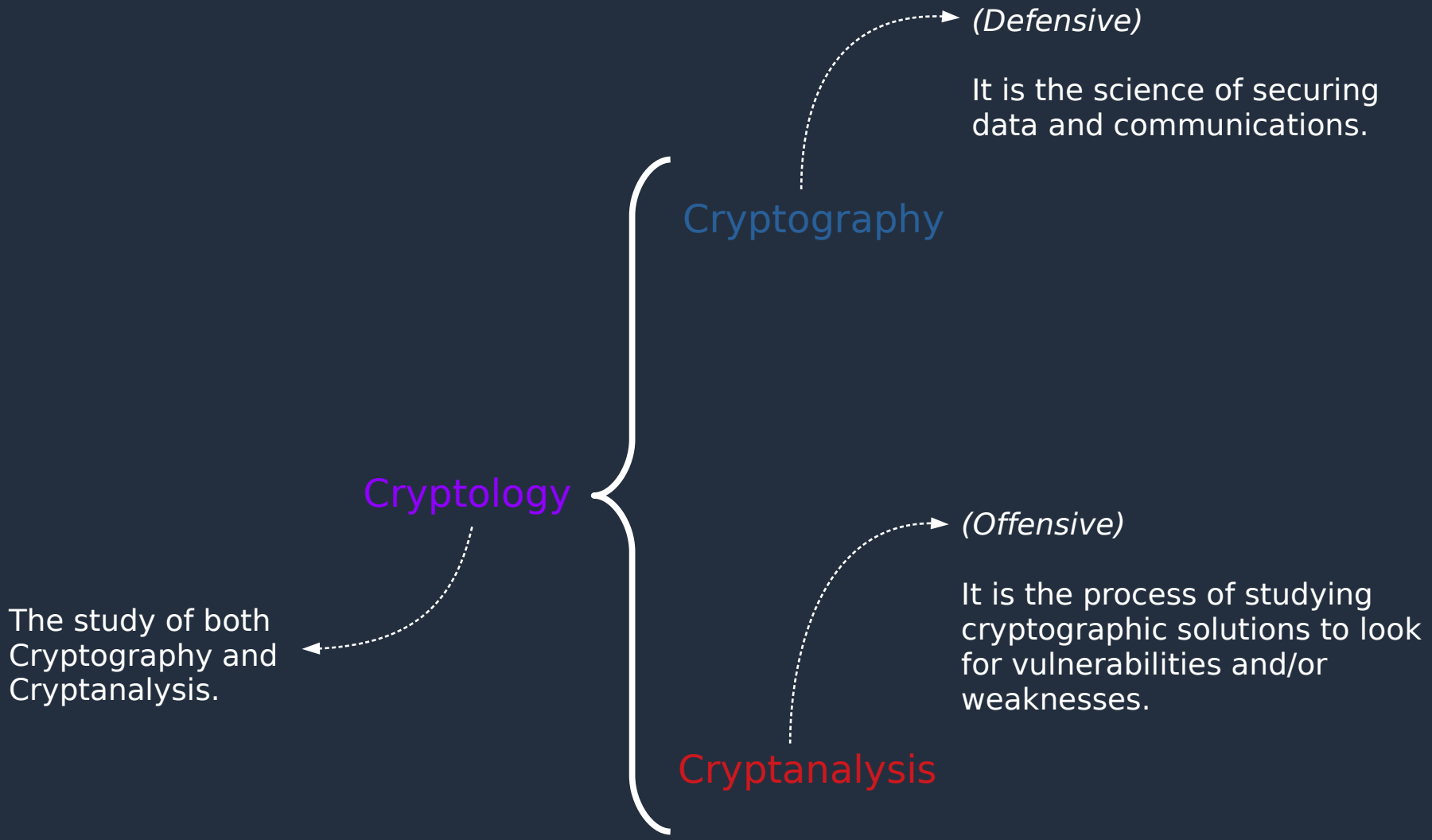
Threat Modeling Methods

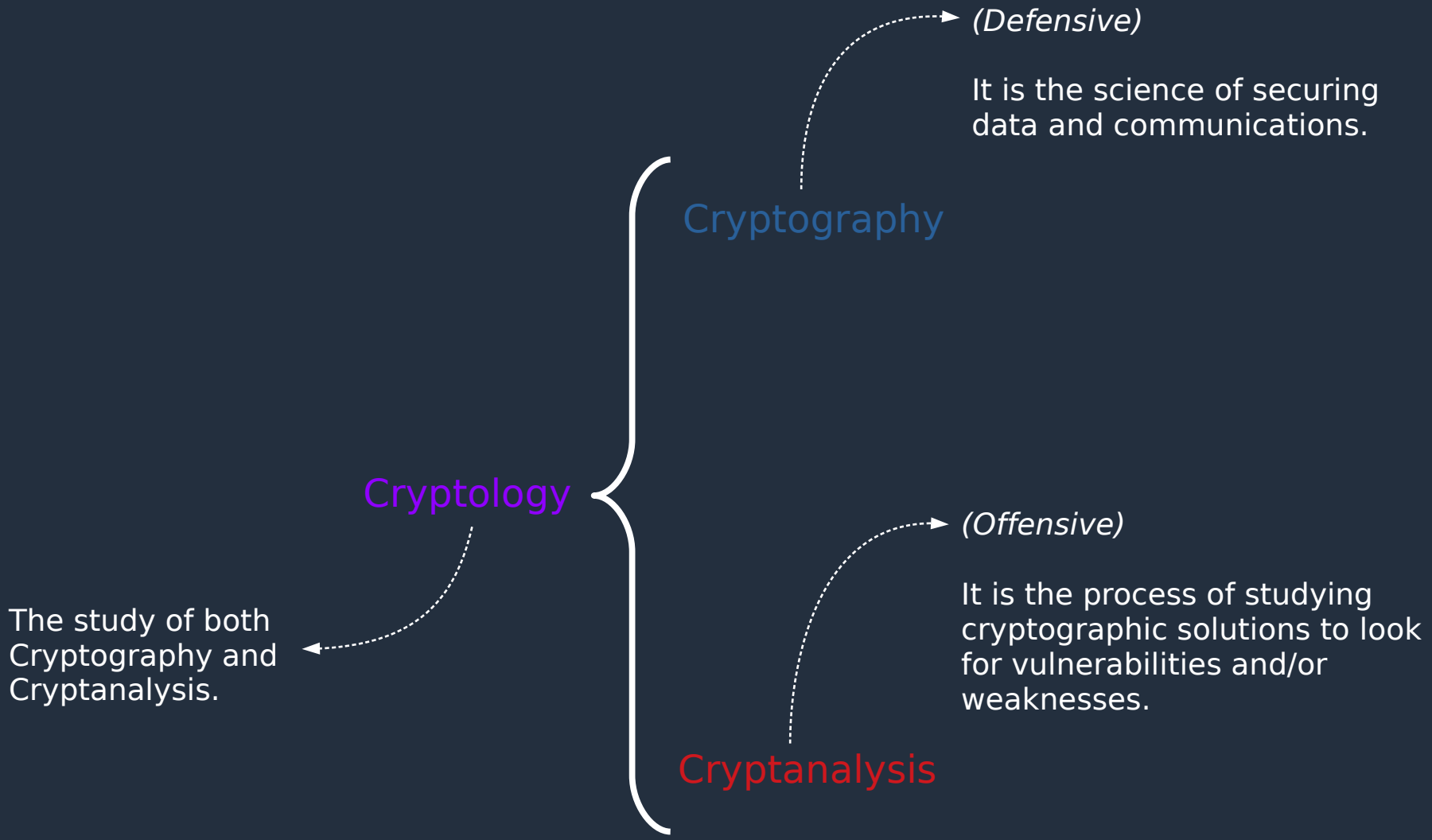
- STRIDE.
- PASTA.
- Trike.
- Attack Trees.
- CRUD.
- Security Cards.
- OCTAVE.
- VAST Modeling.
- ...

There’s no best option.

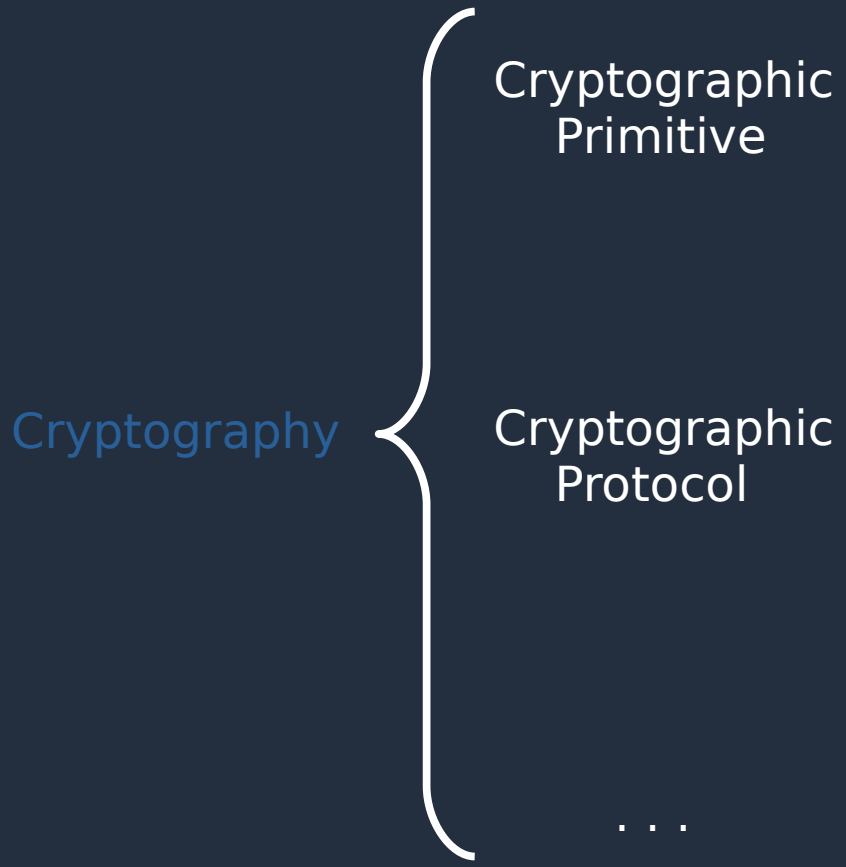
The complexity and completeness of the model are the keys.

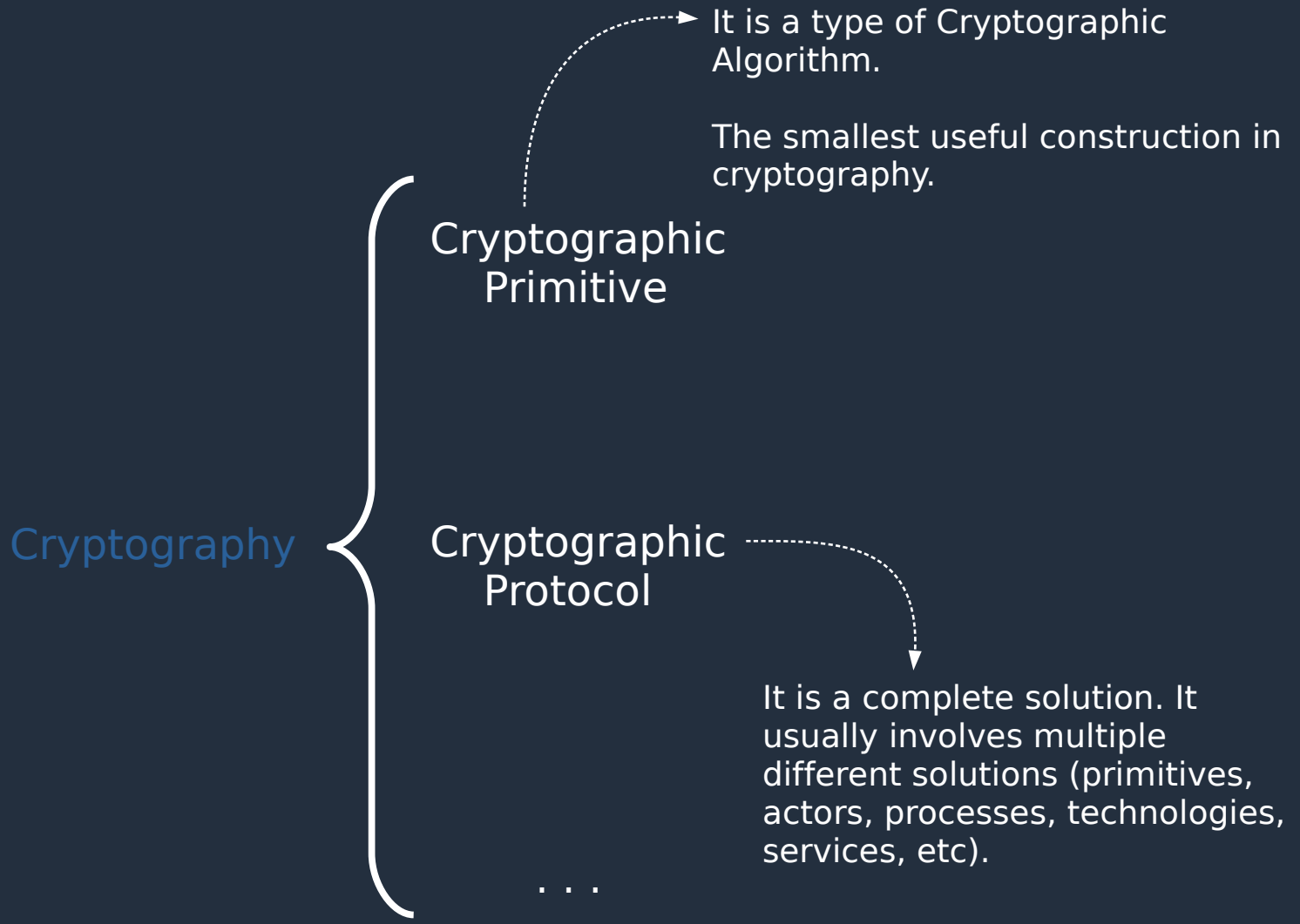

Cryptography
Cryptology
Cryptanalysis

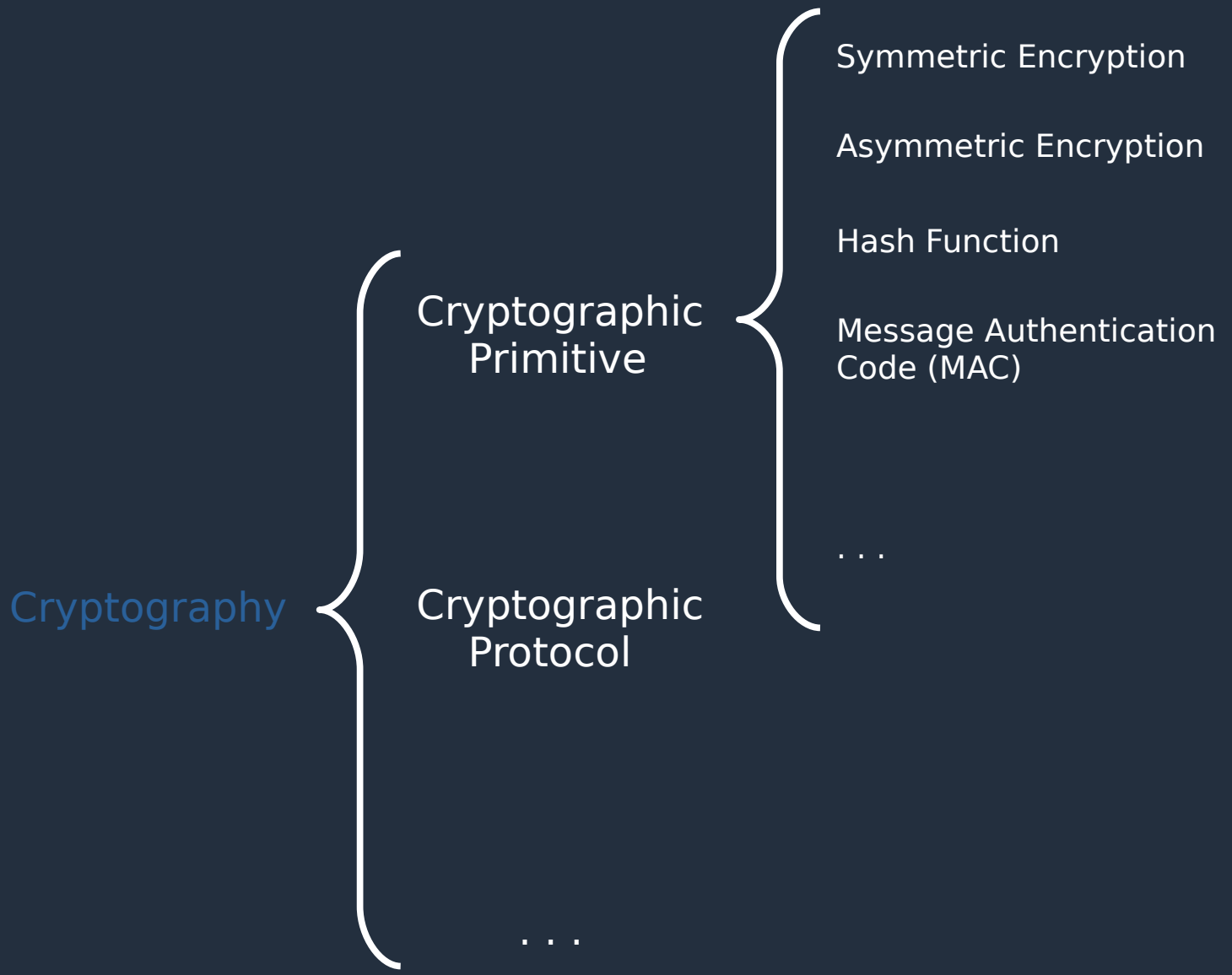

(Defensive)
It is the science of securing data and communications.
Cryptography
Cryptology
The study of both Cryptography and Cryptanalysis.
(Offensive)
It is the process of studying cryptographic solutions to look for vulnerabilities and/or weaknesses.
Cryptanalysis



Kerckhoff's principle: "only the key is kept secret".


Cryptography
Cryptographic Primitive
Cryptographic Protocol
. . .


It is a type of Cryptographic Algorithm.
The smallest useful construction in cryptography.
Cryptographic Primitive
Cryptography
Cryptographic Protocol
It is a complete solution. It usually involves multiple different solutions (primitives, actors, processes, technologies, services, etc).
. . .


Cryptography
Cryptographic Primitive
Cryptographic Protocol
. . .
Symmetric Encryption
Asymmetric Encryption
Hash Function
Message Authentication Code (MAC)
. . .

“The world runs on C code. While other languages may offer newer language features, their compilers and libraries are typically written in C”.

Robert C. Seacord
(Secure Coding Gran Master)

Process

Pointer dereference.

Threat

Dereference an invalid address.

Mitigation

- Strategy 1: ...
- Strategy 2: ...

Process

Recursive function.

Threat

Infinite recursion.

Mitigation

- Strategy 1: ...
- Strategy 2: ...

Process

Recursive function.

Threat

Infinite recursion.

It is not specific to C, right.

Mitigation

- Strategy 1: ...
- Strategy 2: ...

Process

Loop.

Threat

Infinite Loop.

Mitigation

- Strategy 1: ...
- Strategy 2: ...

malloc() ⇢ adapter__malloc()

```
adapter__malloc()
{
        regular adaptation, plus:

        malloc();
        initialization();
        casting();
}
```

free() ⇢ adapter__free()

```
adapter__free()
{
        regular adaptation, plus:

        adapter__memset();
        free();
        pointer = NULL;
}
```

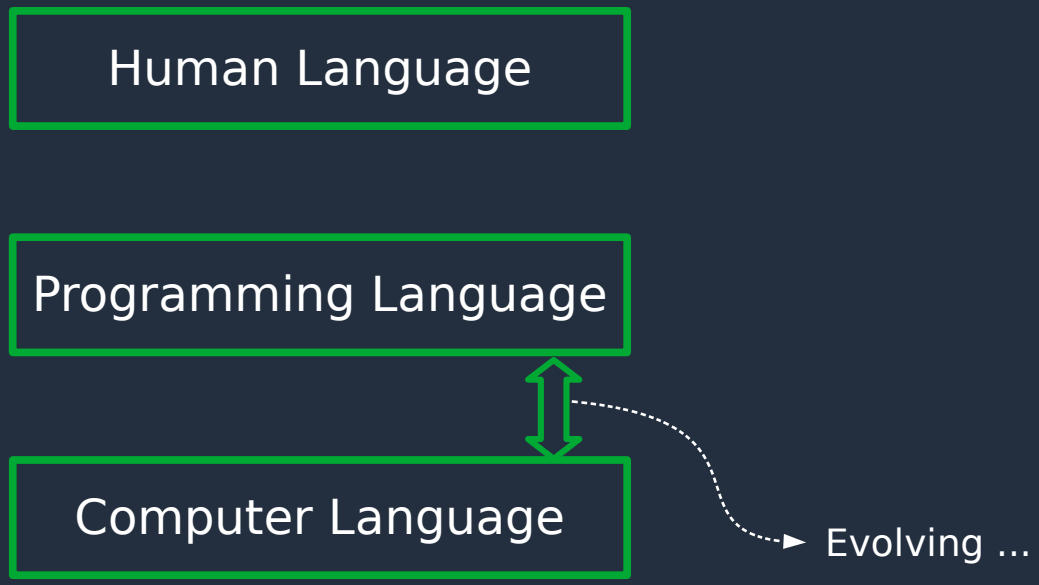
memset() ⇢ adapter__memset()

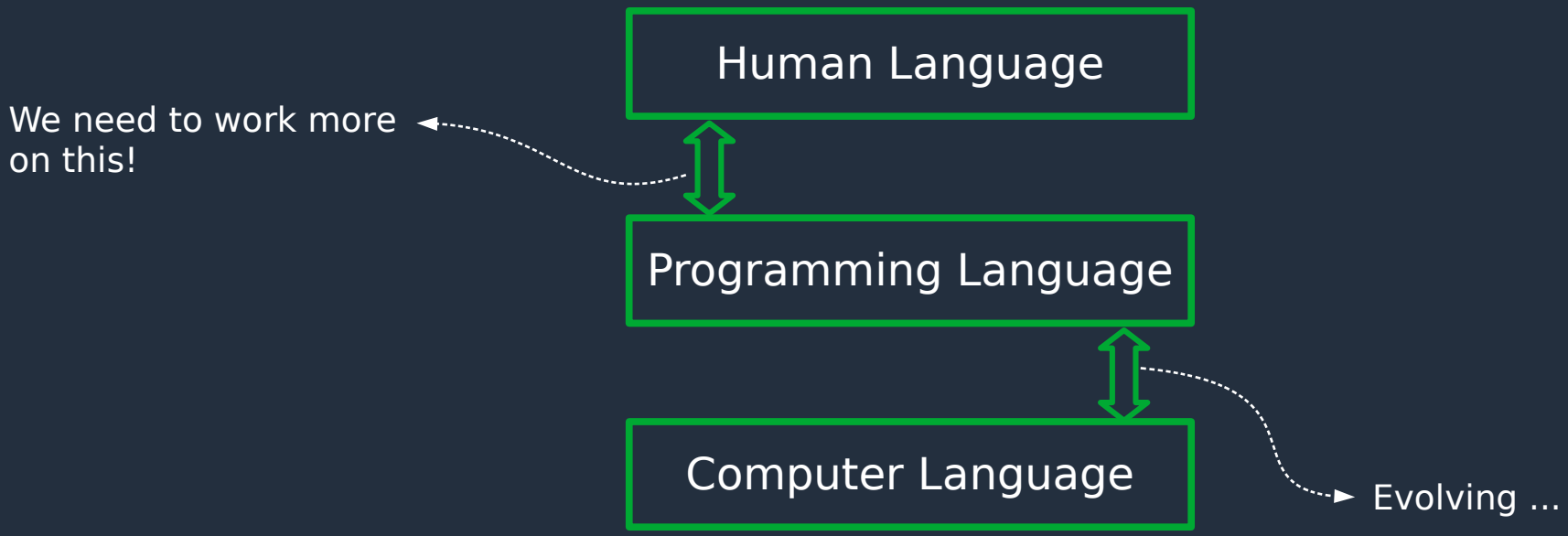
```
adapter__memset()
{
        regular adaptation
}
```

Human Language

Programming Language

Computer Language


Human Language
Programming Language
Computer Language
Evolving ...


Human Language
We need to work more on this!
Programming Language
Computer Language
Evolving ...

Keep it simple.