RUB

Behind Closed Curtains

Insights on Security Vulnerabilities in Smartphone Basebands

Daniel Klischies

SECURITY AND PRIVACY
OF UBIQUITOUS SYSTEMS

Smartphones Under Attack

Forbes

FORBES > INNOVATION > CYBERSECURITY

New Pixel 'Spyware' Warning—Google Deletes 'Dangerous' App On Millions Of Phones

Zak Doffman Contributor

Zak Doffman writes about security, surveillance and privacy.

Follow

Aug 15, 2024, 09:00am EDT

Updated Aug 15, 2024, 02:17pm EDT

Recorded Future®

Book a **free** demo

Research (Insikt)

"Mobile NotPetya": Spyware Zero-Click Exploit Development Increases Threat of Wormable Mobile Malware

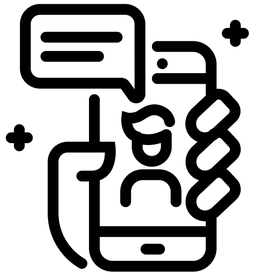
npr

DONATE

EUROPE

French Media Report President Macron's Cellphone Was A Spyware Target

Defenses



App Level

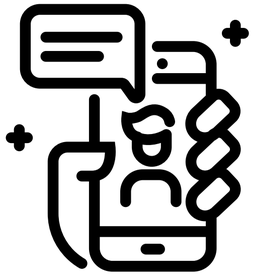
Sandboxing
→ Isolation from other apps and the system

Permission Model
→ Limits access to system and other apps

Use of Memory Safe Languages
→ Prevents buffer overflows, use-after-free…

And many more…

Defenses

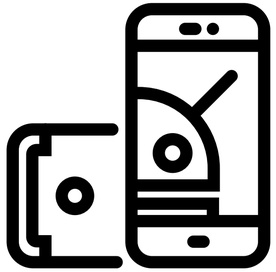


App Level

Sandboxing
→ Isolation from other apps and the system

Permission Model
→ Limits access to system and other apps

Use of Memory Safe Languages
→ Prevents buffer overflows, use-after-free…



System Level

Verified Boot
→ Ensures the system has not been modified

Address Space Layout Randomization
→ Makes exploitation more difficult

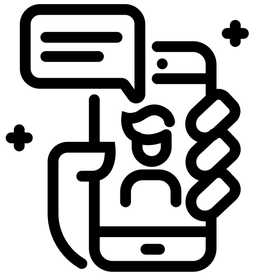
Kernel Control Flow Integrity
→ Makes exploitation more difficult

And many more…

“Classic” Infection Path

Threat Actor

First Stage
(Malicious web link,
WhatsApp message…)



Compromise App
(Browser, Messenger…)

→ **Remote Code Execution**

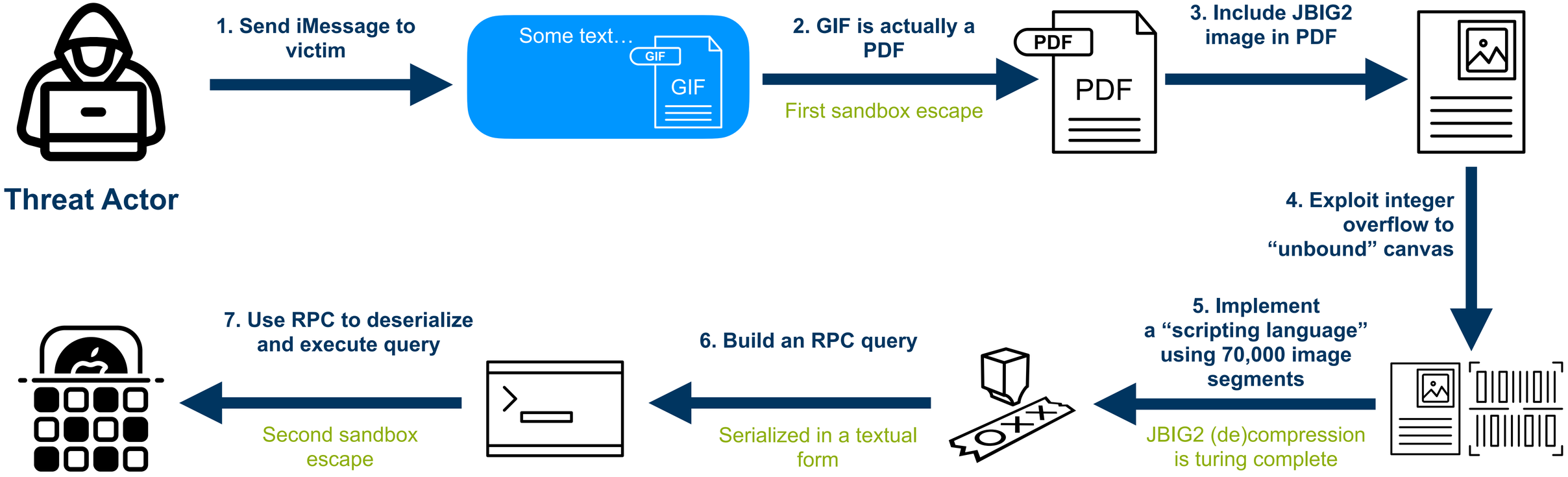
Second Stage
(System-level exploit)

Compromise System
(“become root”)

→ **Privilege Escalation**

Increasing exploit chain complexity

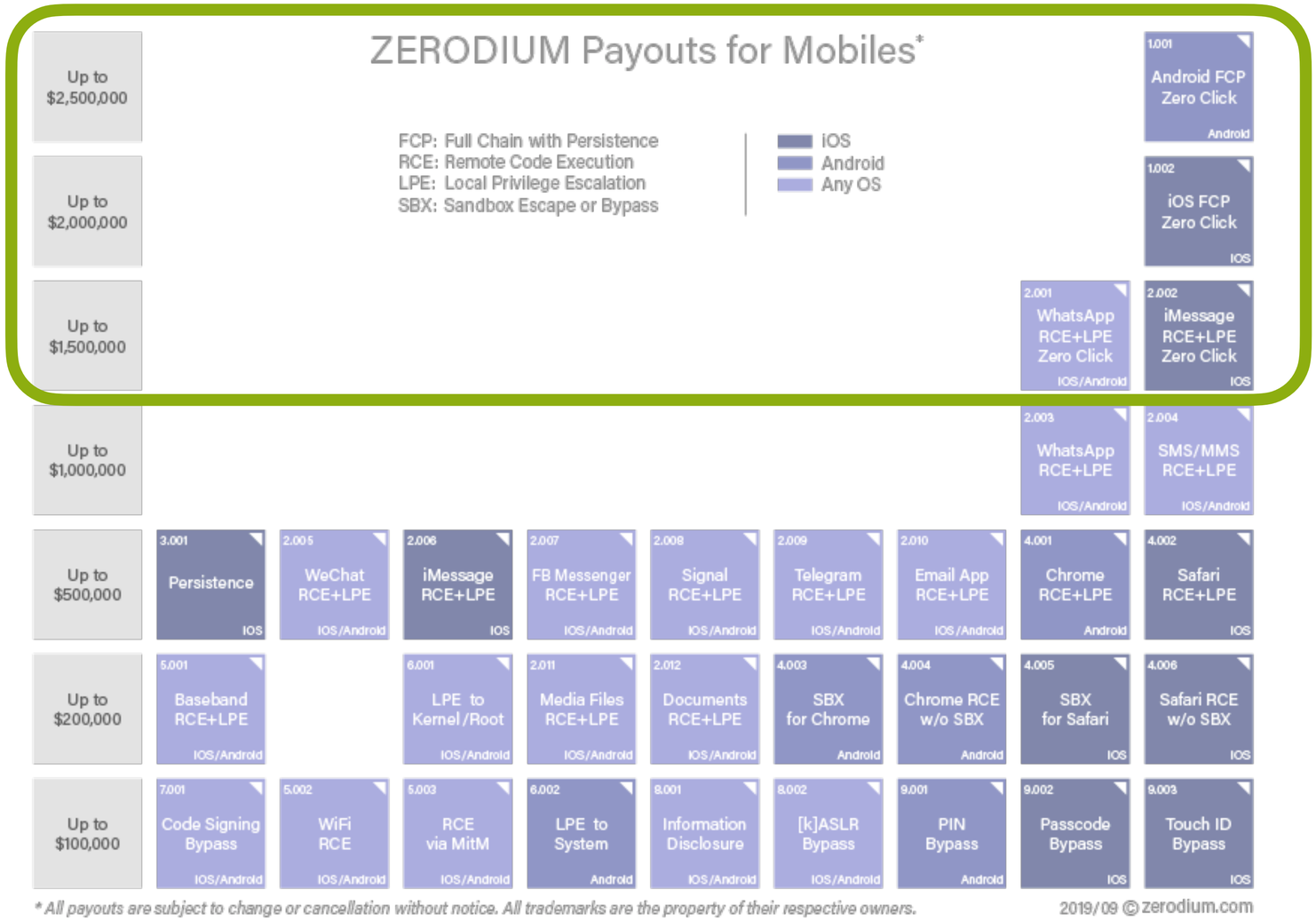
NSO Group's zero-click iMessage exploit FORCEDENTRY



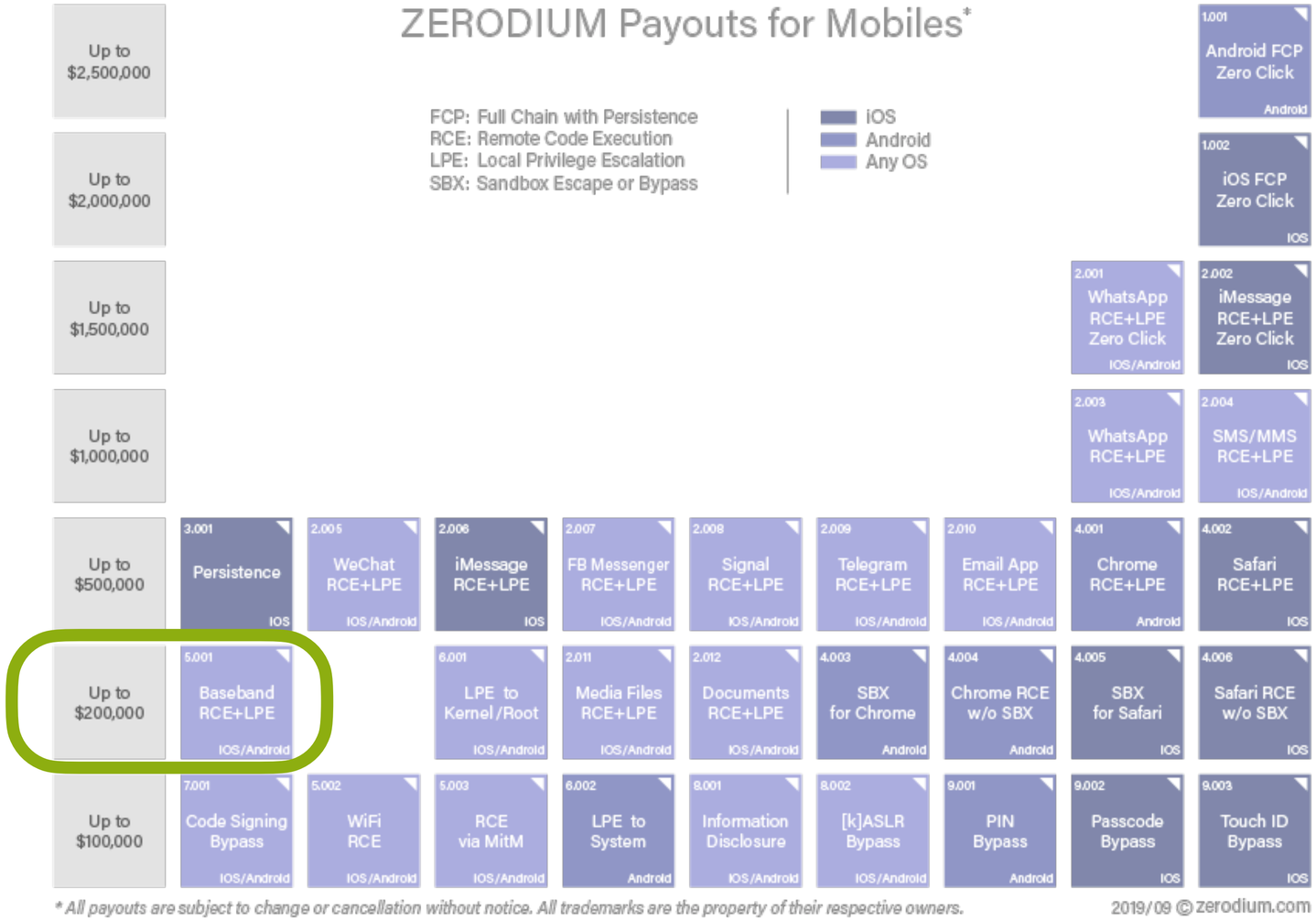
https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html

https://googleprojectzero.blogspot.com/2022/03/forcedentry-sandbox-escape.html

Exploit chains are rare and expensive…



Baseband exploits are a cheaper alternative…



Basebands and the Predator spyware

Intellexa's Predator spyware was used to target phones of politicians, ambassadors and journalists around the world

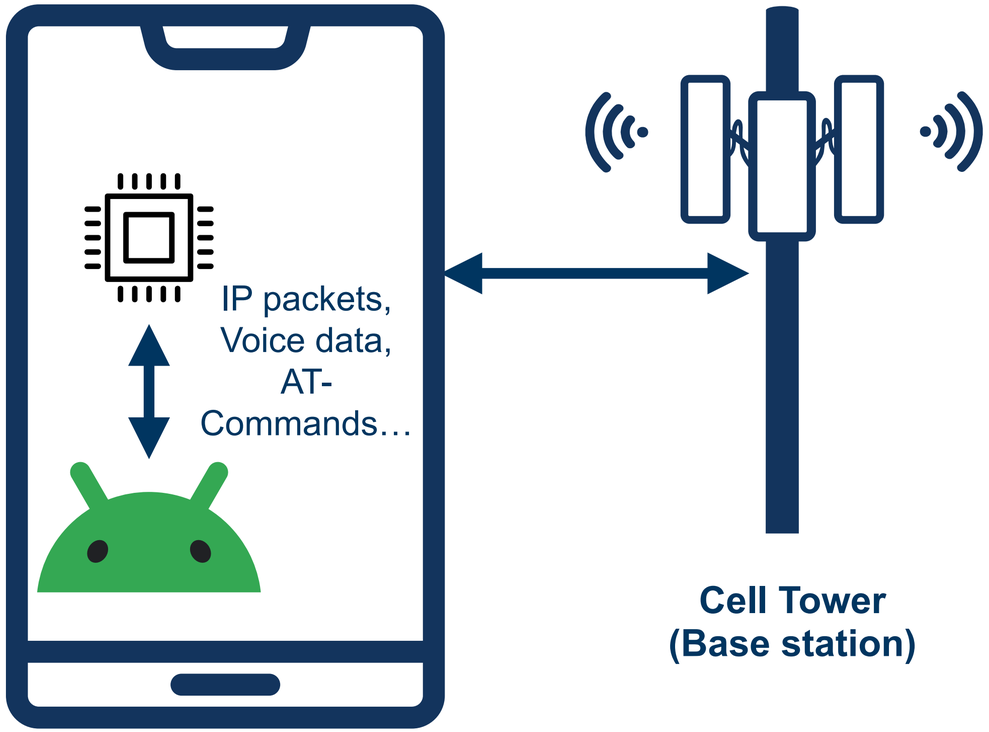
In 2023, sales presentations on Predator were leaked

One infection vector: Basebands

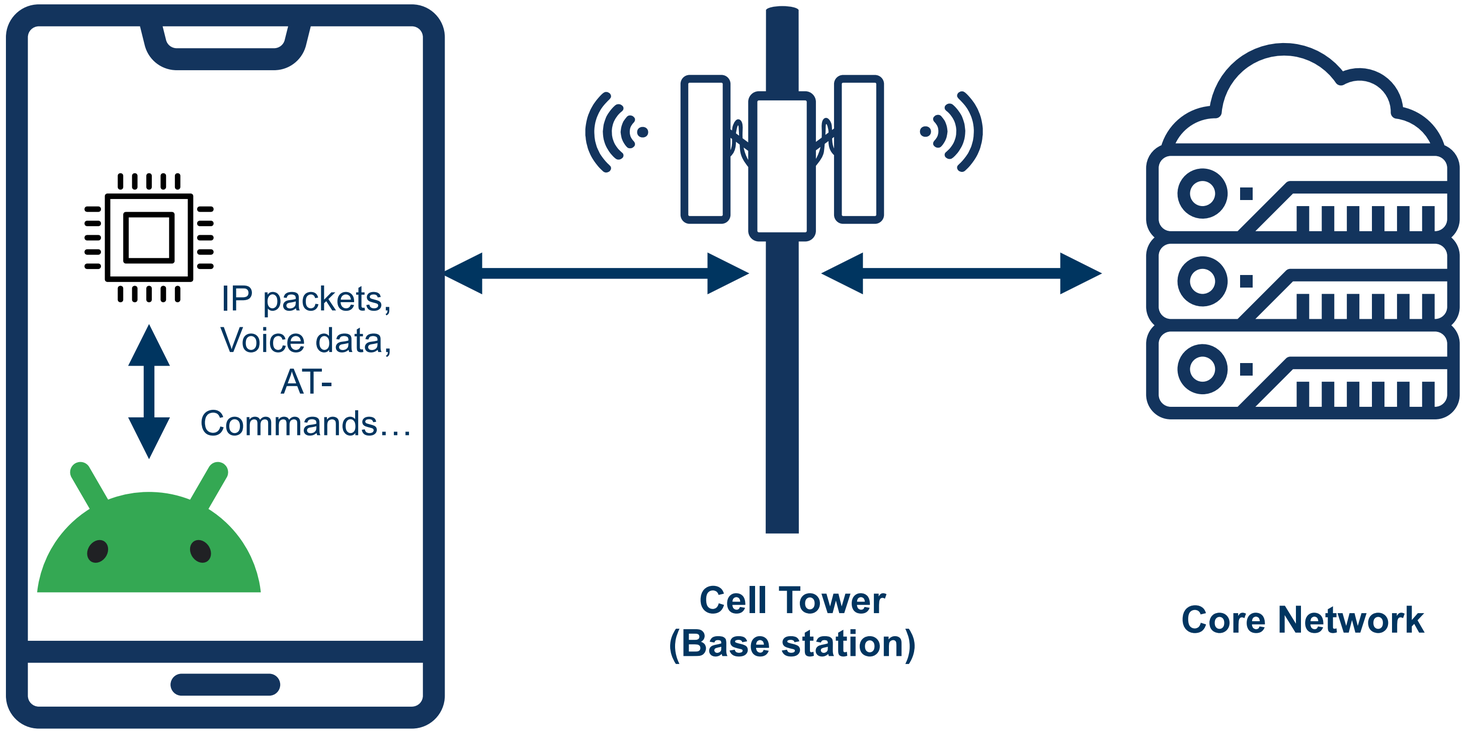
https://securitylab.amnesty.org/latest/2023/10/technical-deep-dive-into-intellexa-alliance-surveillance-products/

Cellular Basebands and Networks

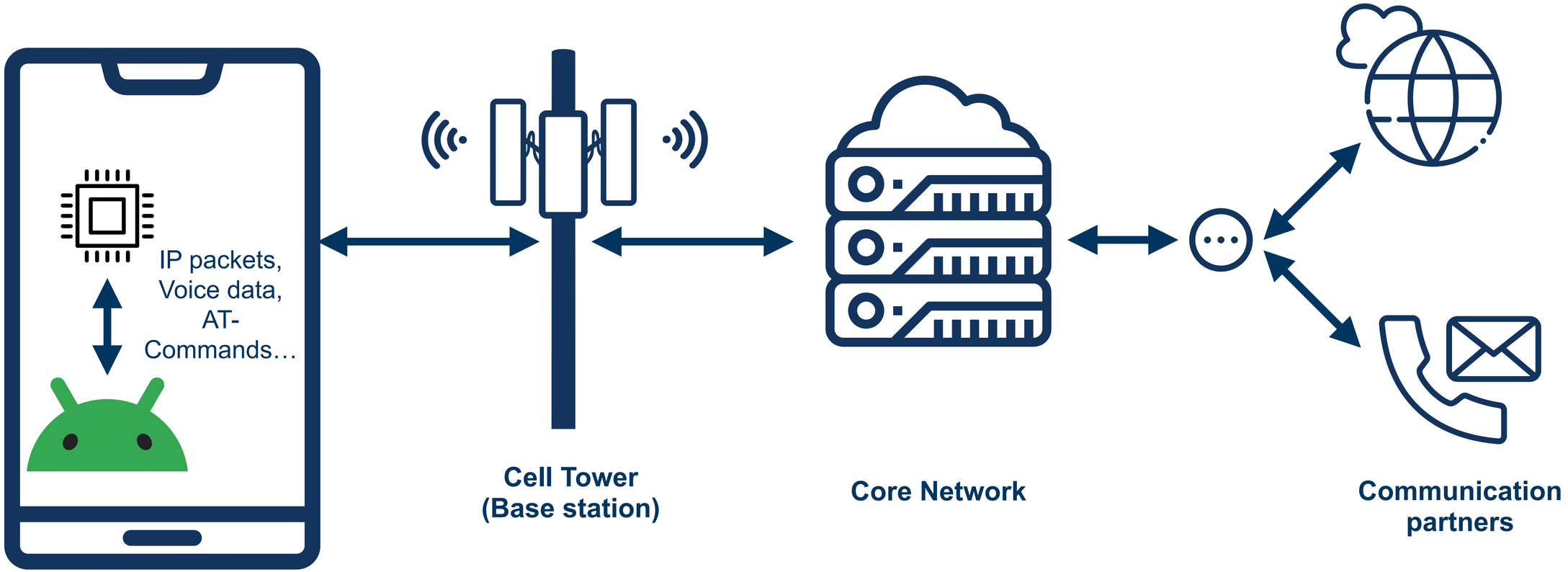
Basebands: Overview



Basebands: Overview



Basebands: Overview



LTE Functionality

User Plane

Voice & SMS data

Internet connectivity

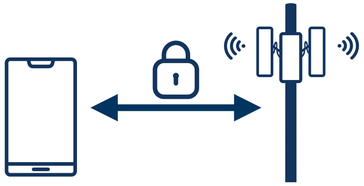
LTE Functionality

User Plane

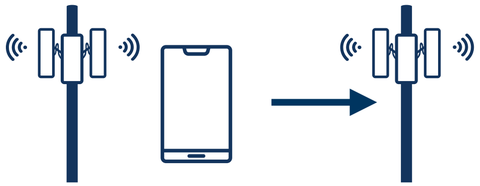
Voice & SMS data

Internet connectivity

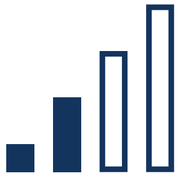
Control Plane



Authentication / Authorization



Mobility Management



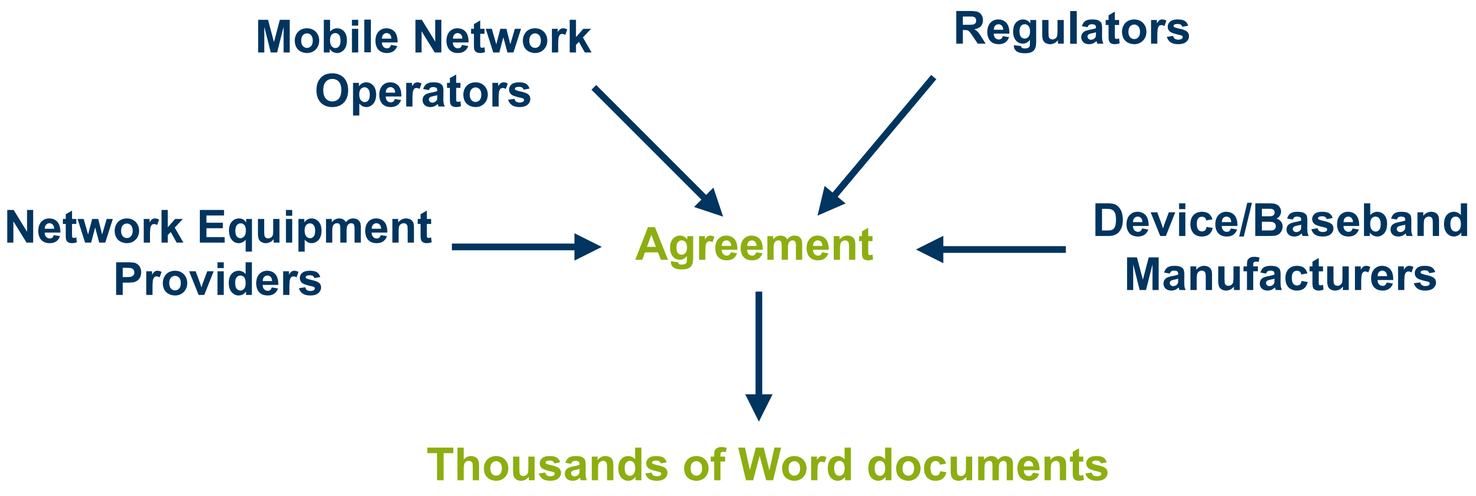
QoS

Call setup

Radio Resource Control

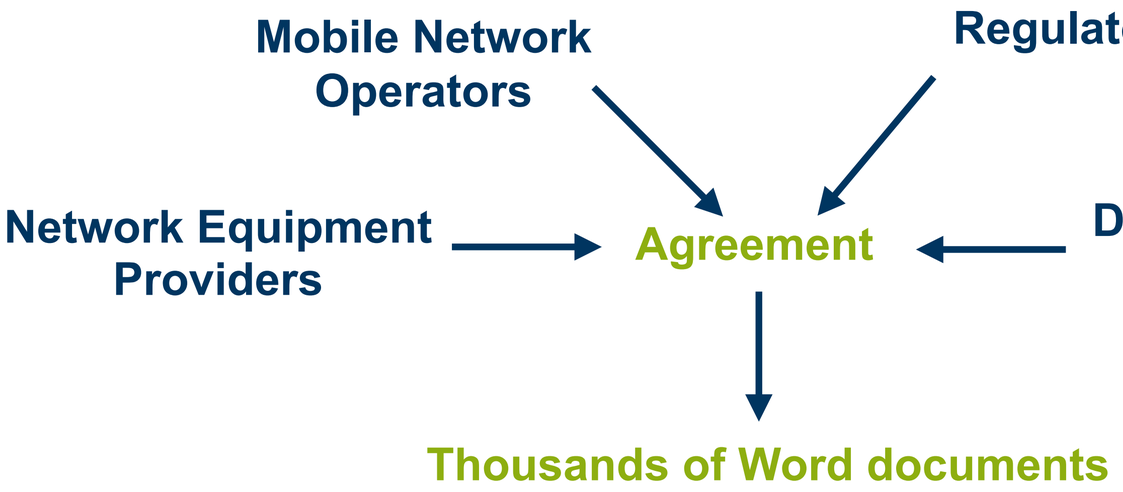
3GPP Specifications

How do you ensure that every cellular baseband is compatible with every base station and mobile network? → Specifications and Standards



3GPP Specifications

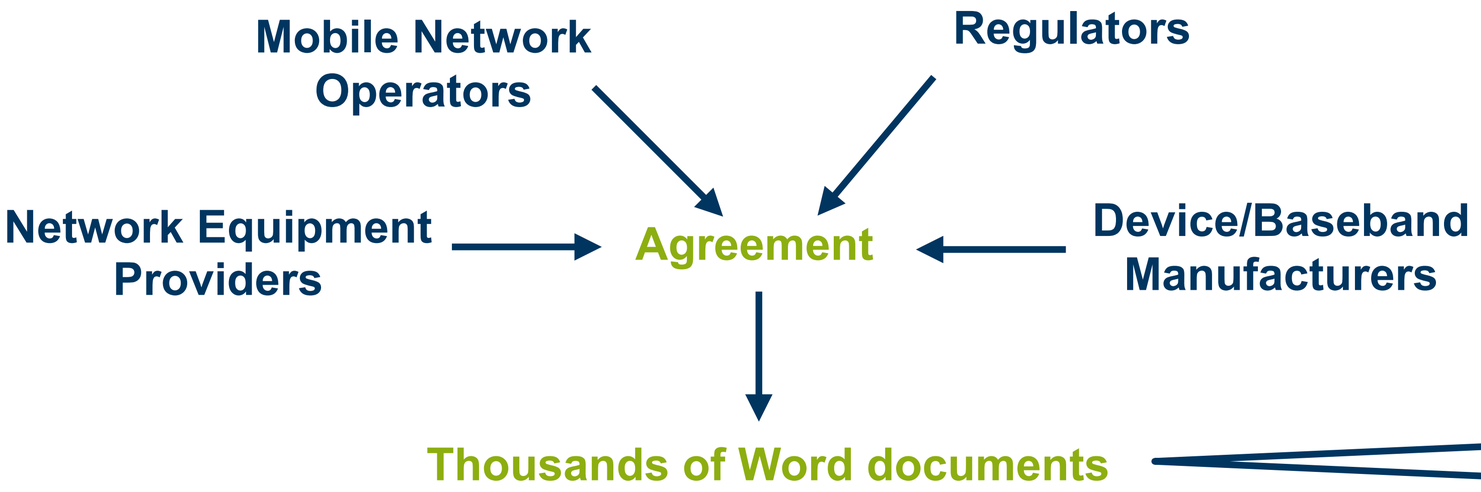
How do you ensure that every cellular baseband is compatible with every base station and mobile network? → Specifications and Standards



Business requirements:
Customer requests, reusability, patents

3GPP Specifications

How do you ensure that every cellular baseband is compatible with every base station and mobile network? → Specifications and Standards

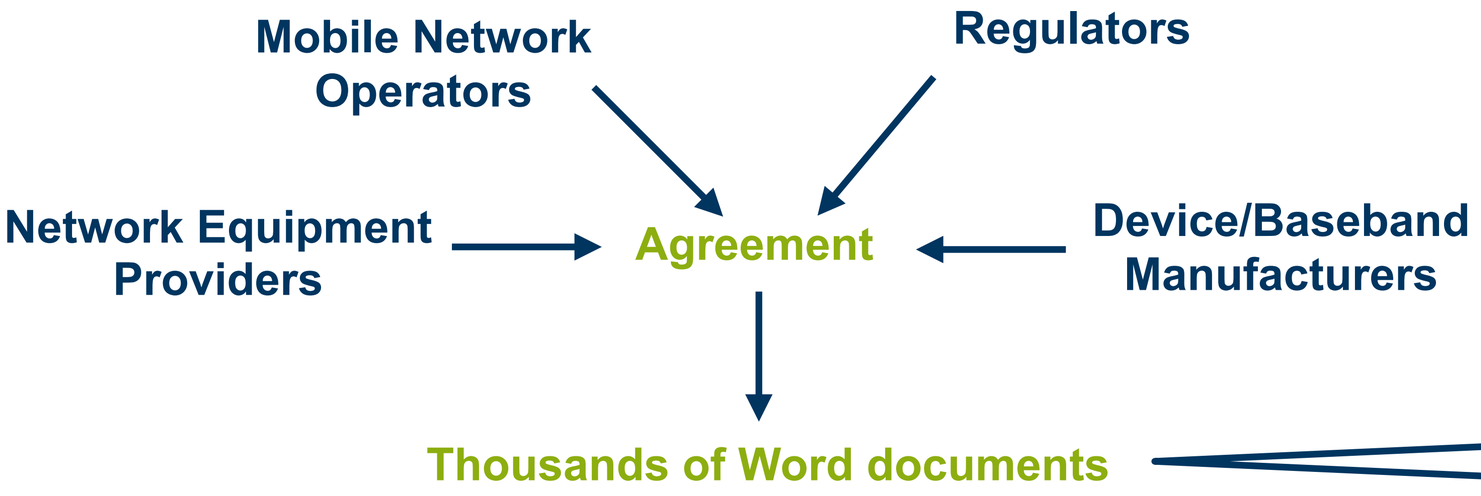


Business requirements: Customer requests, reusability, patents

Goals: Interoperability, Security, Performance…

3GPP Specifications

How do you ensure that every cellular baseband is compatible with every base station and mobile network? → Specifications and Standards



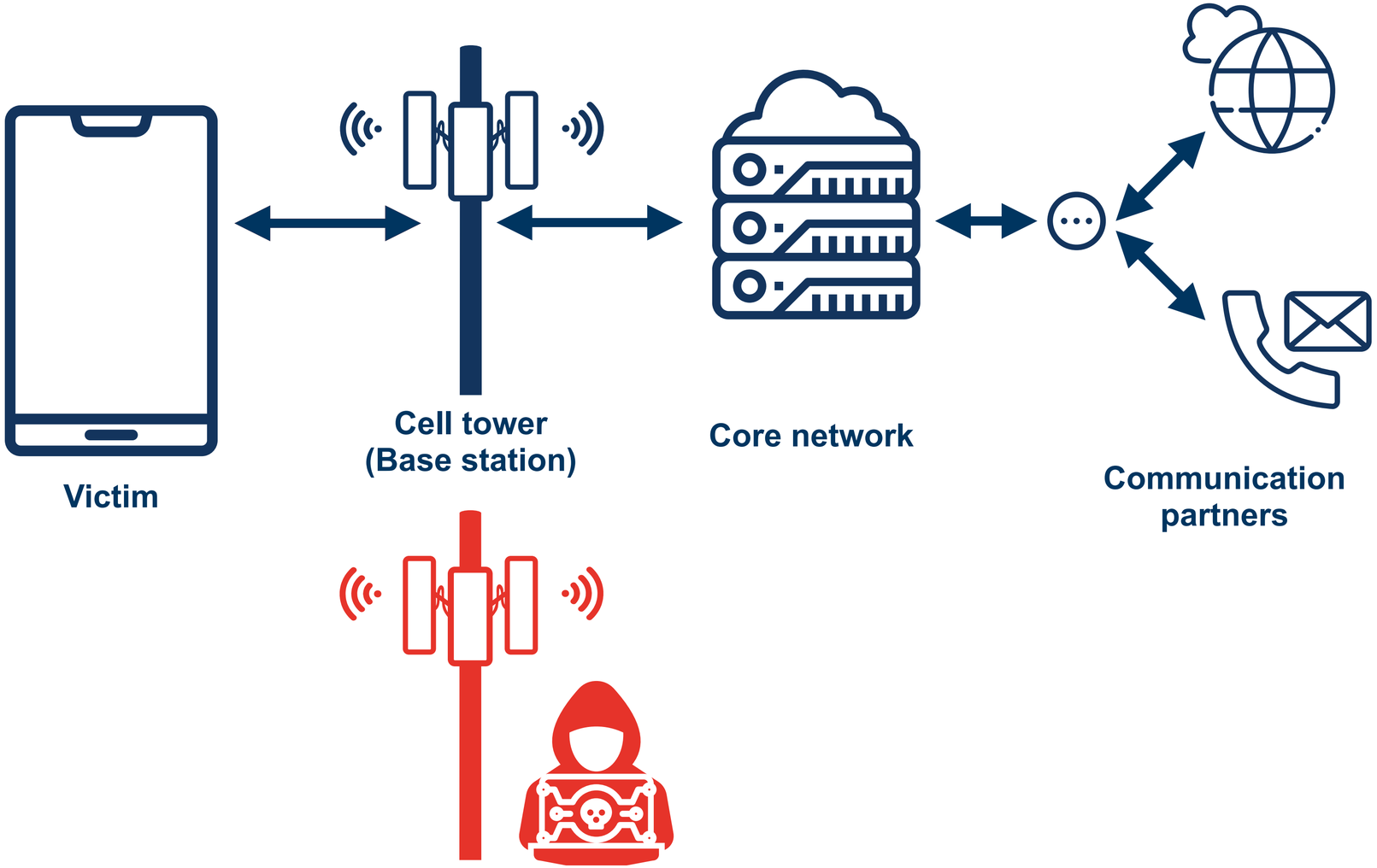
Business requirements: Customer requests, reusability, patents

Goals: Interoperability, Security, Performance…

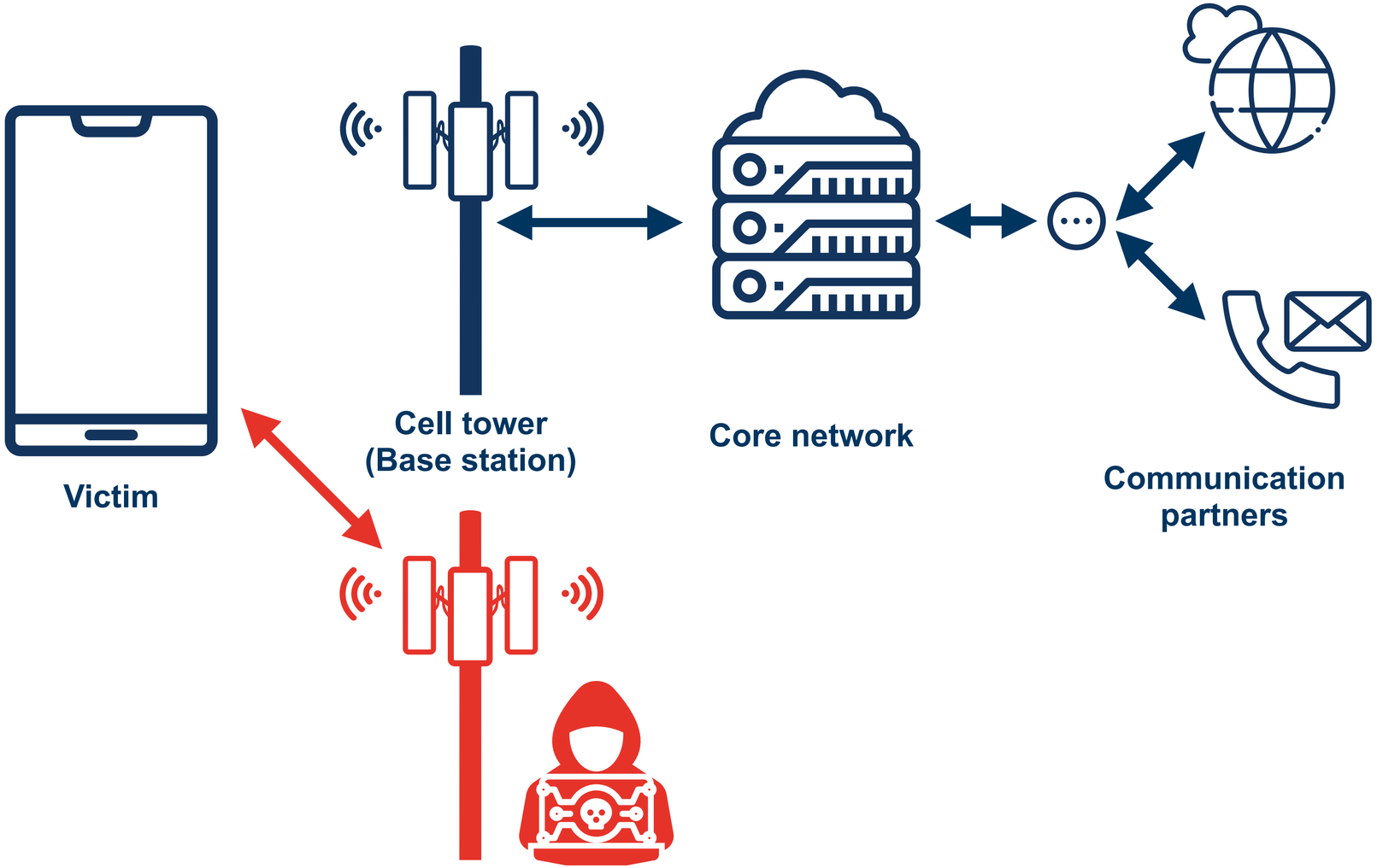
Non-Goals: Completeness, Simplicity

Attacker Models

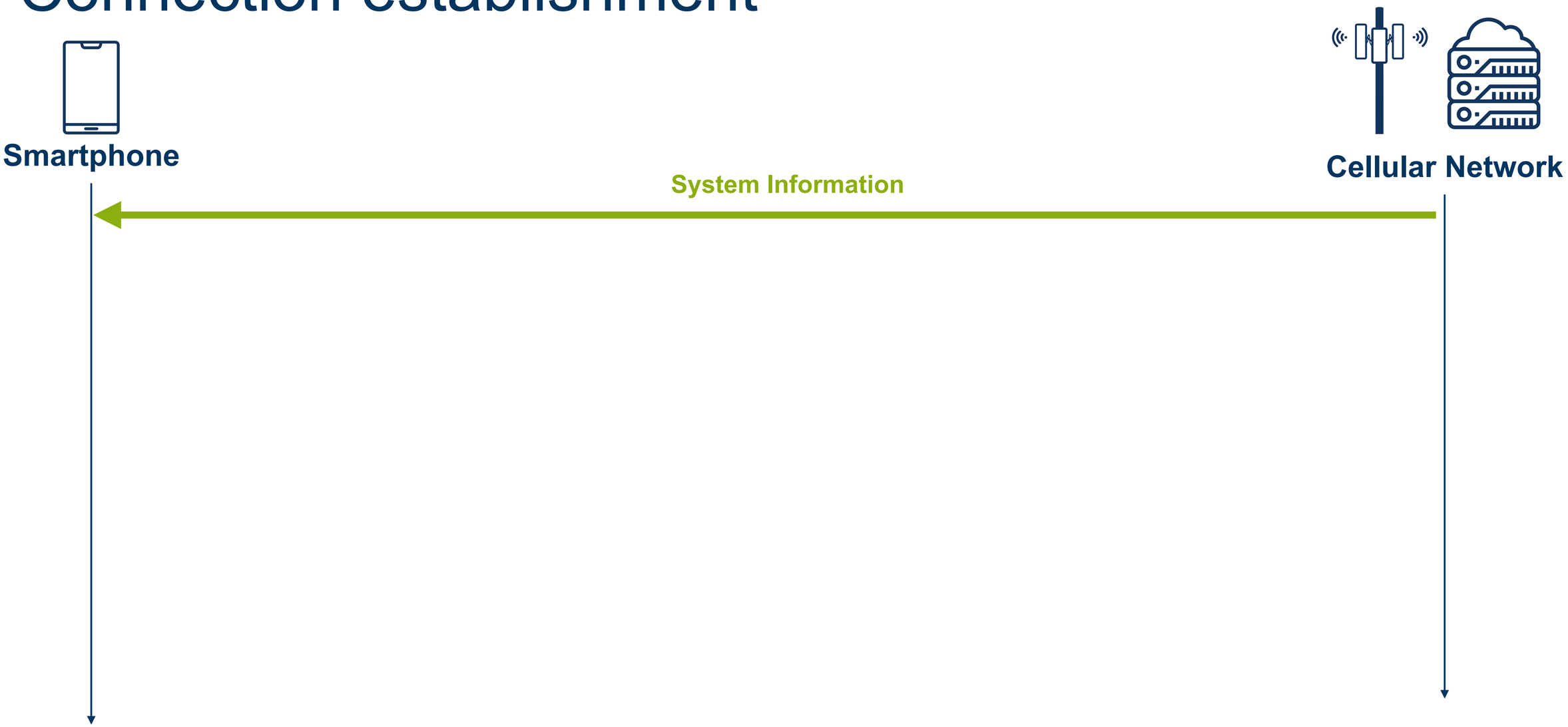
Fake Base Station



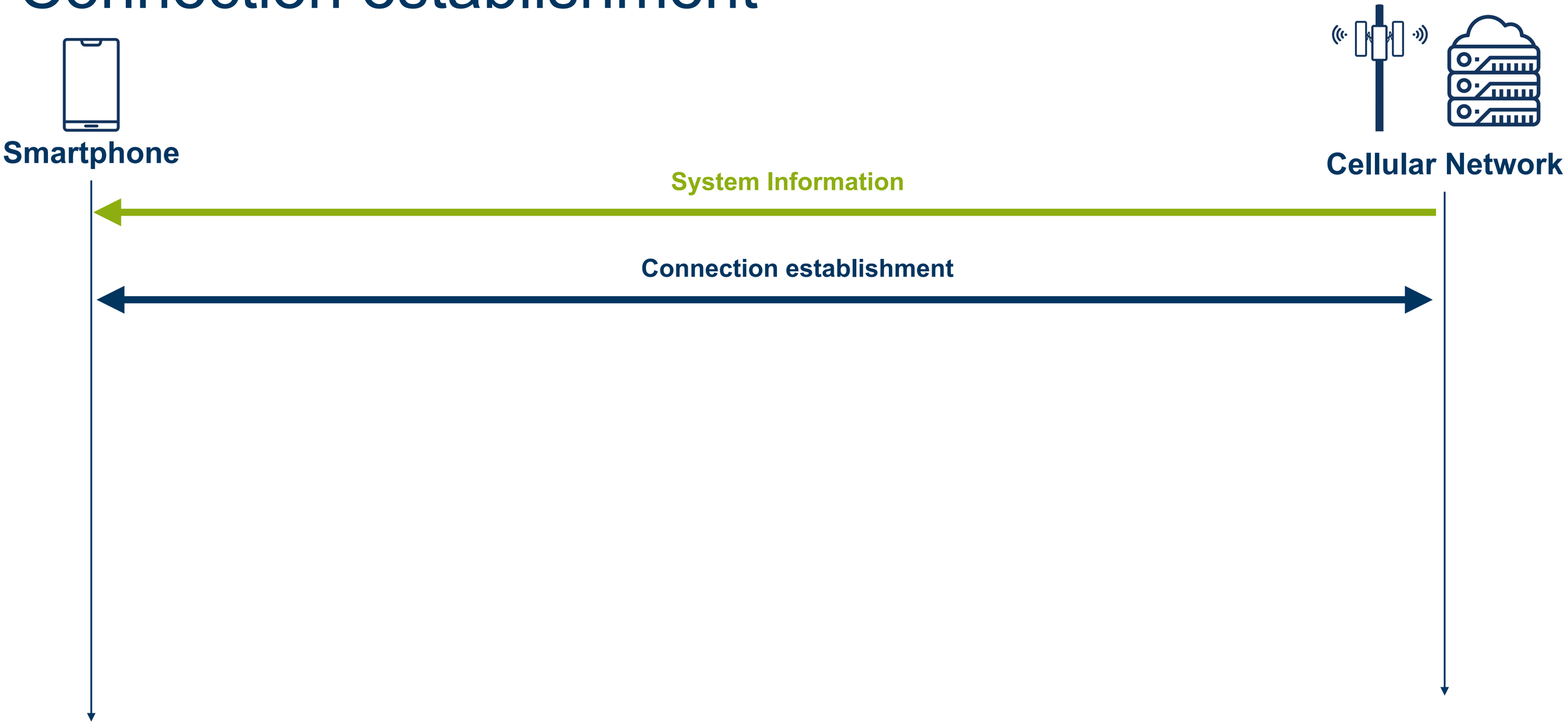
Fake Base Station



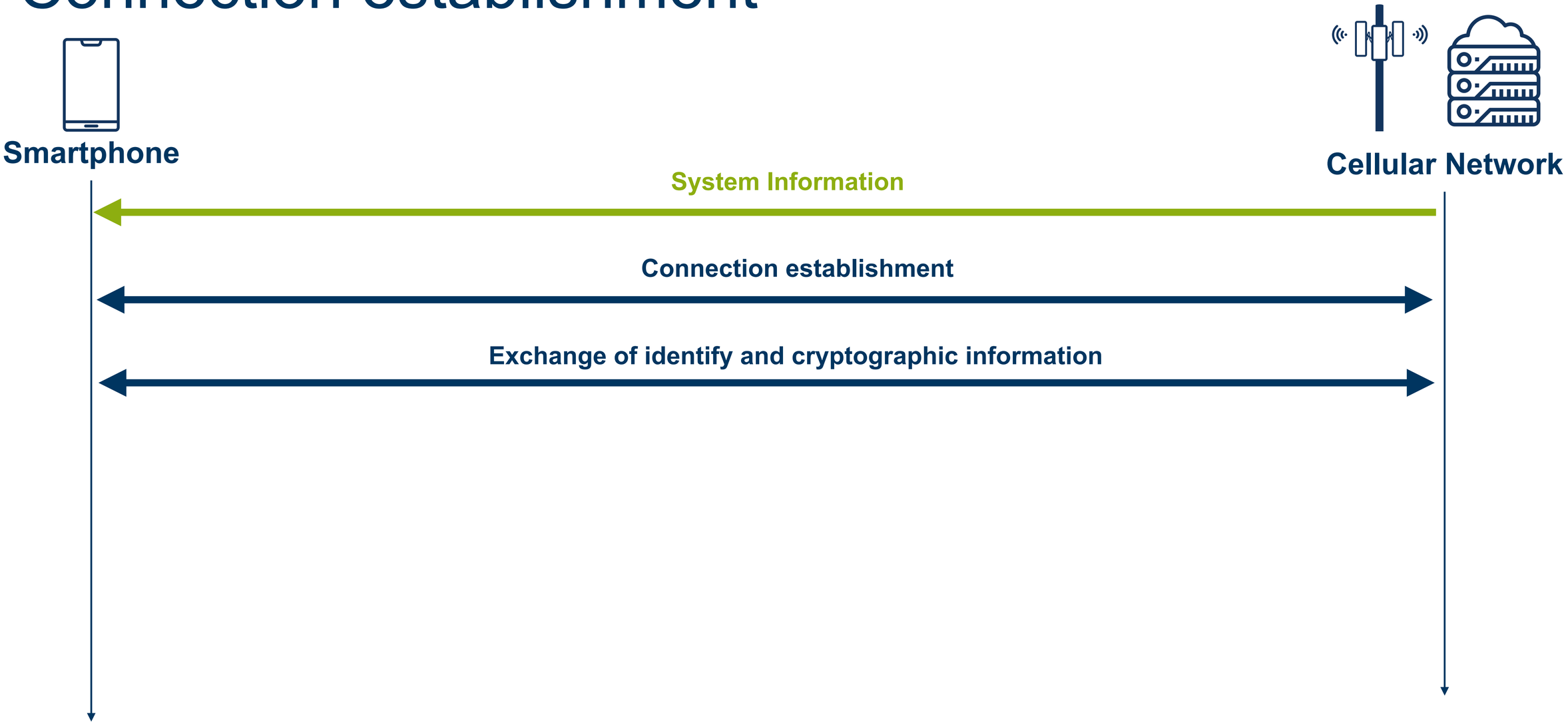
Connection establishment



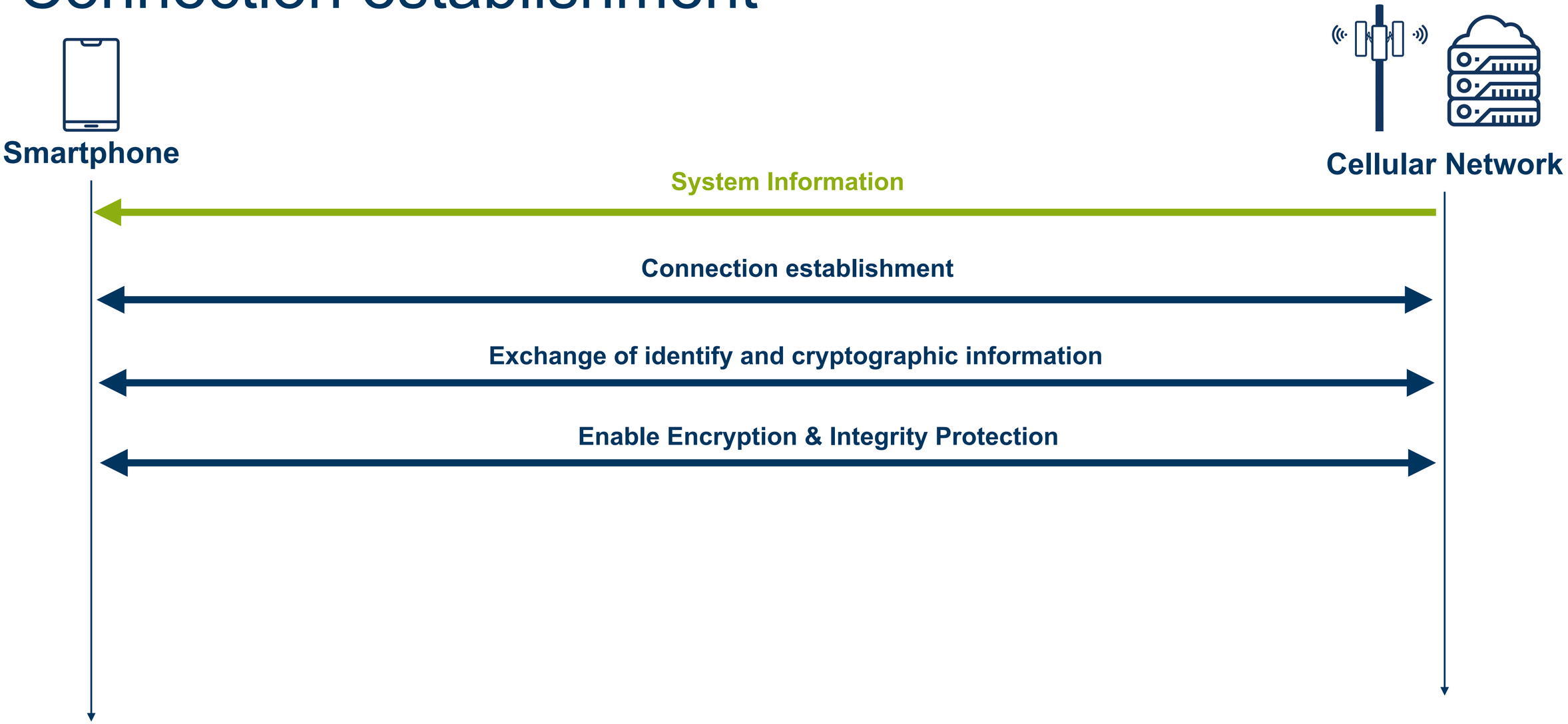
Connection establishment



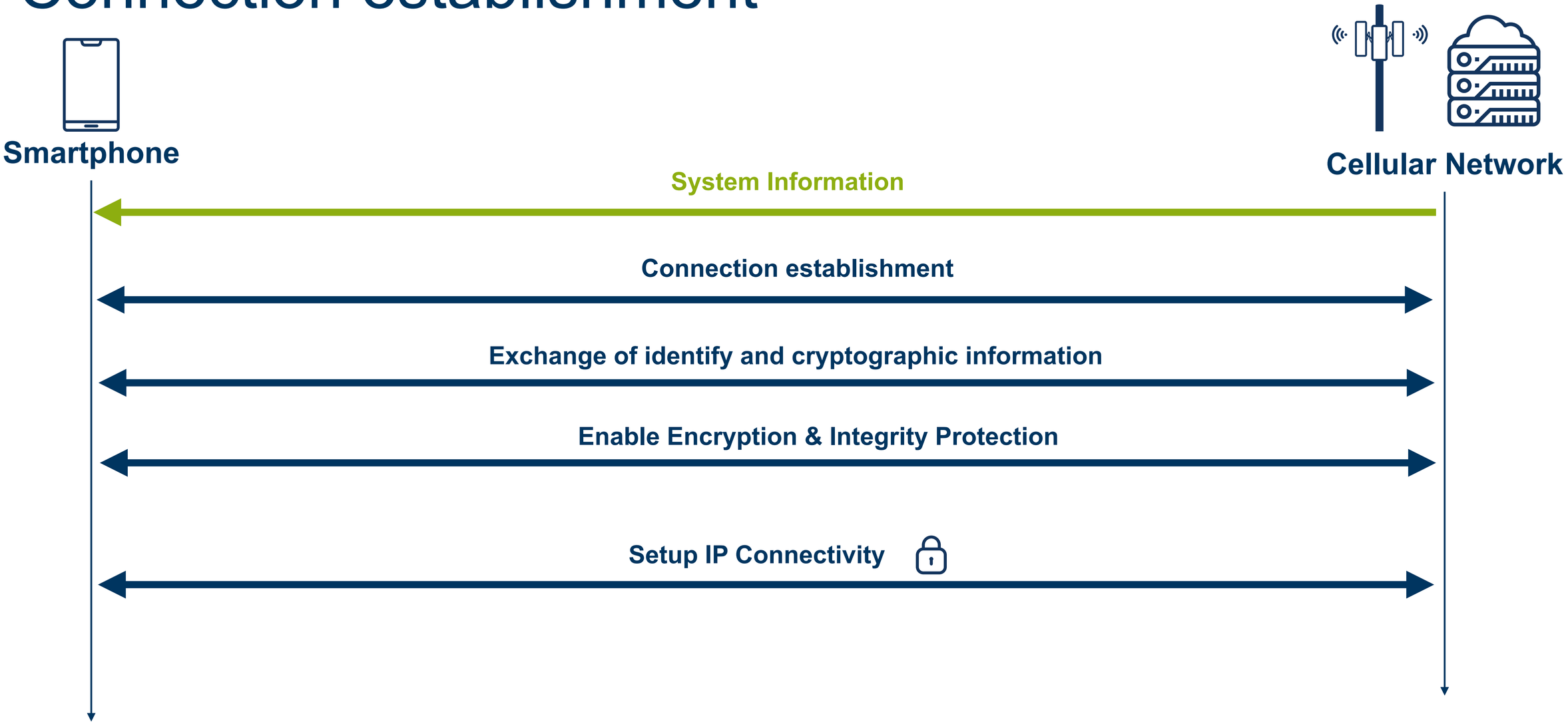
Connection establishment



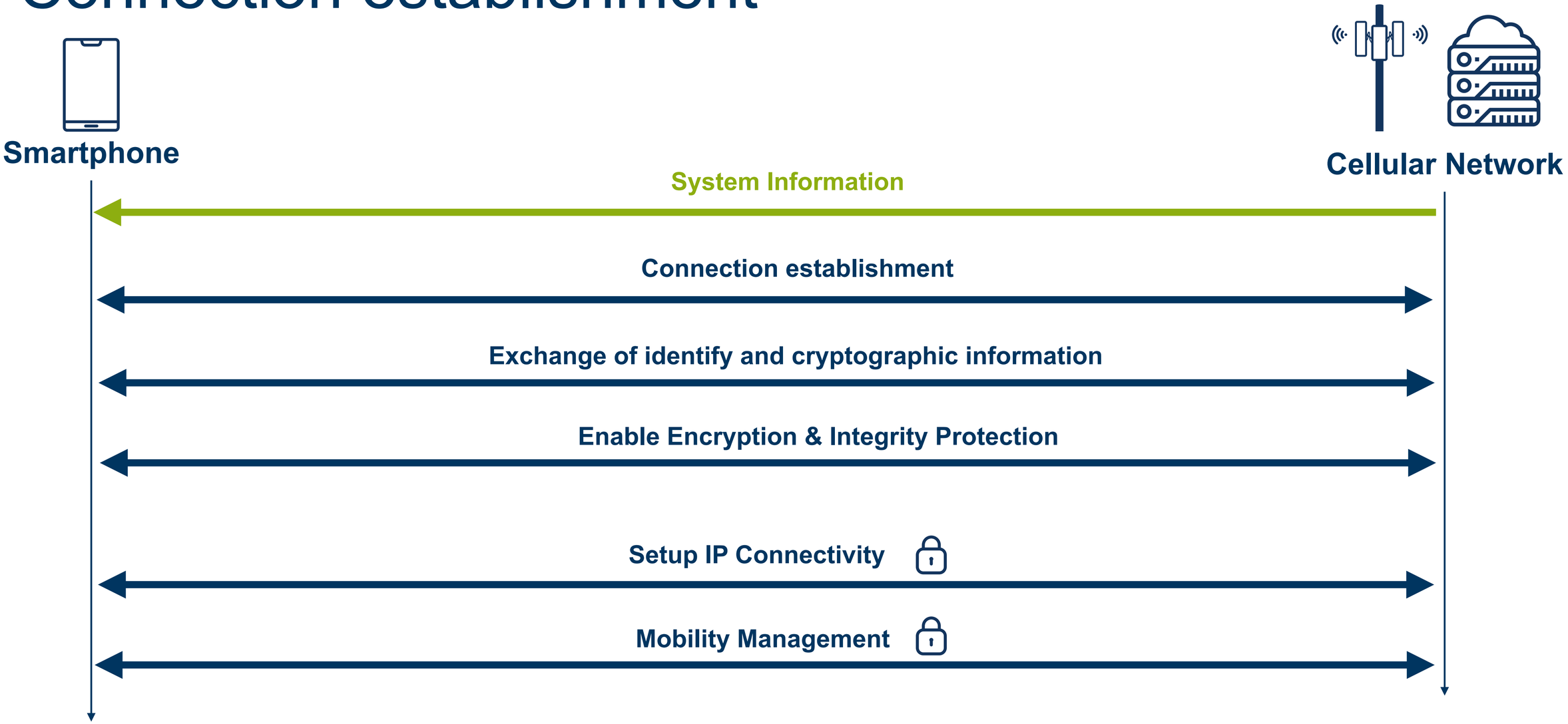
Connection establishment



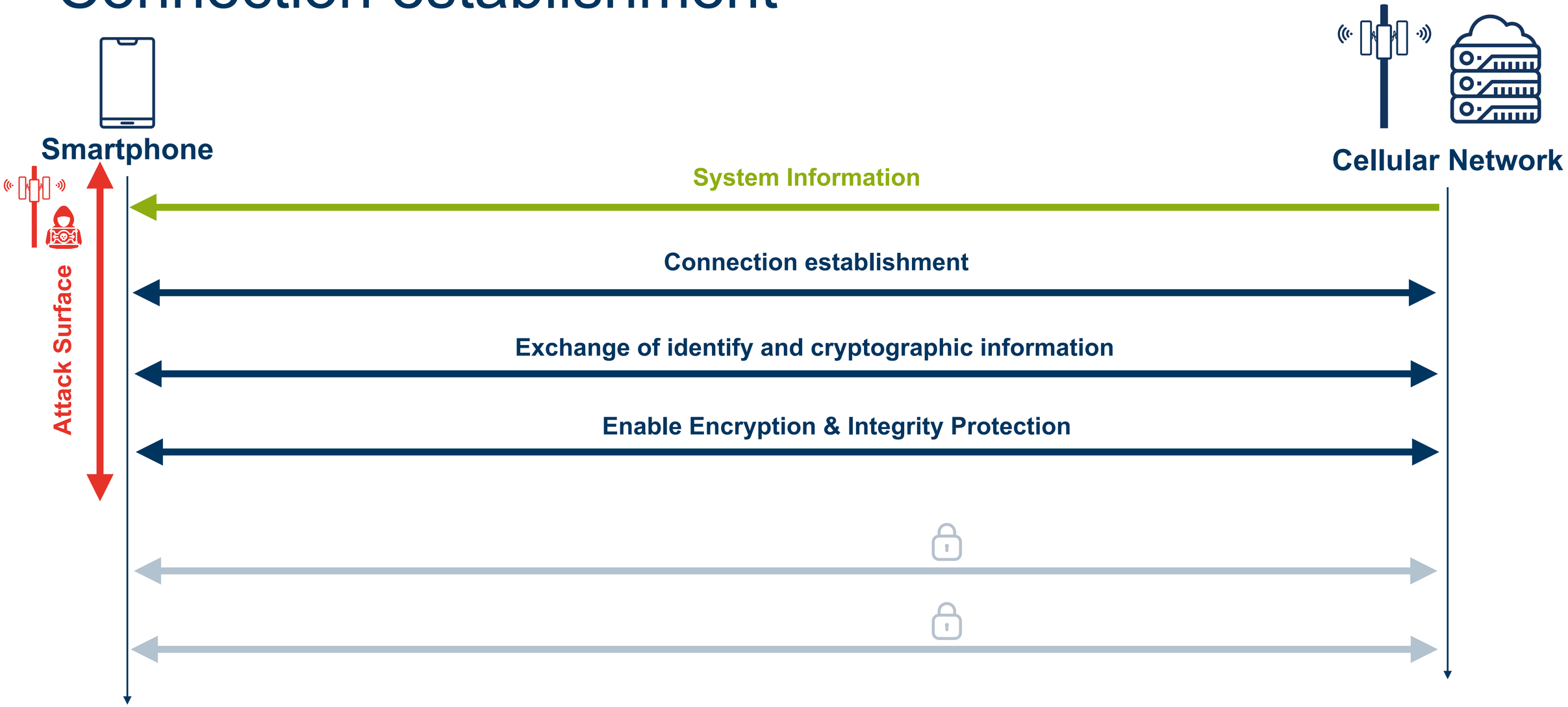
Connection establishment



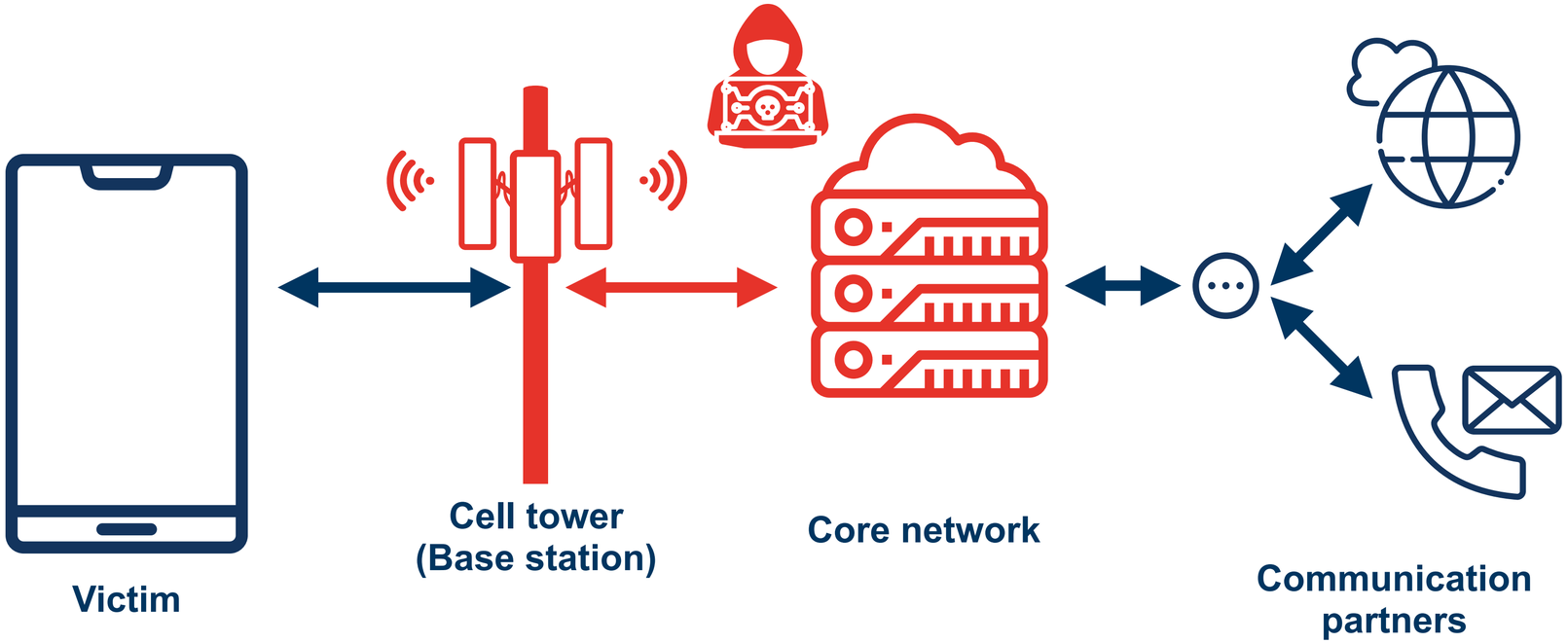
Connection establishment



Connection establishment



Malicious Mobile Network

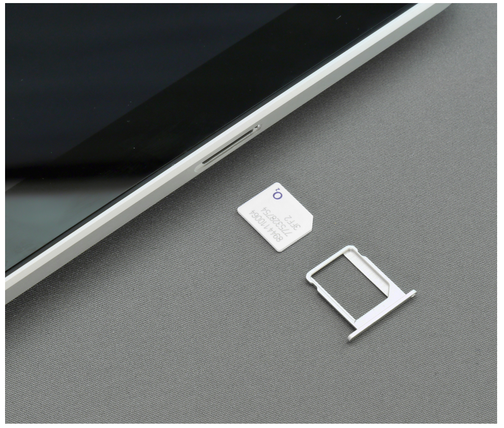


Malicious Mobile Network

Insider

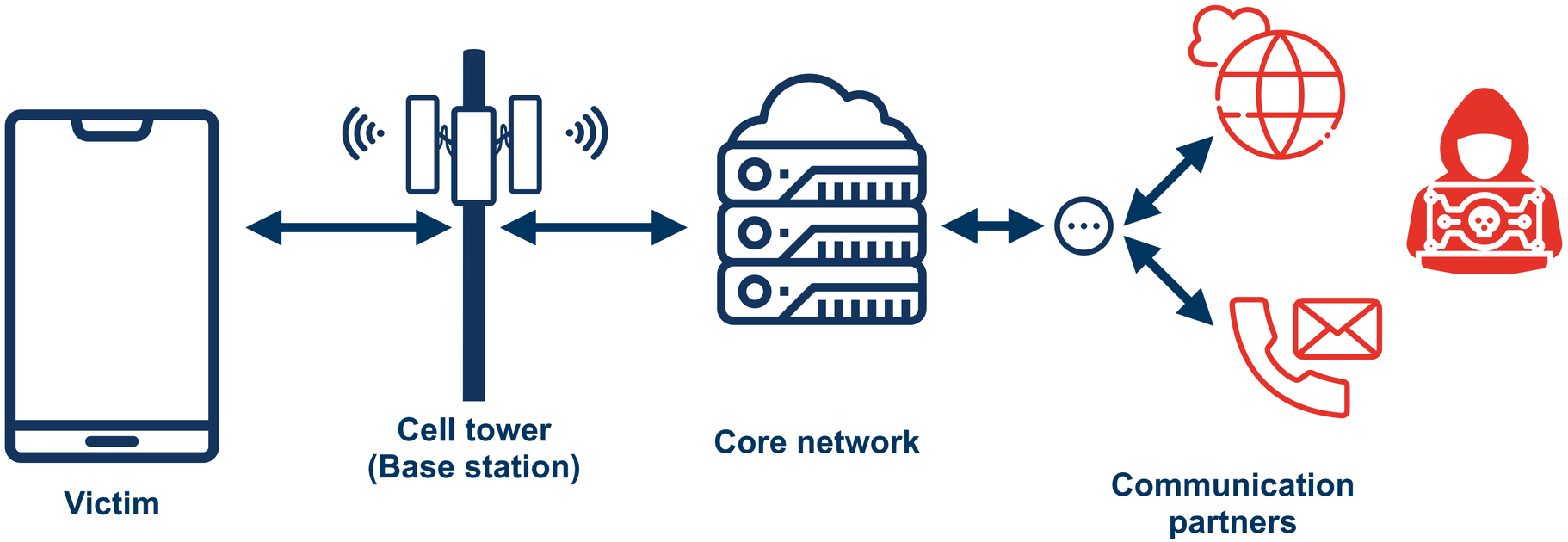
Nation-State actor

Roaming



Physical access

Malicious Communication Partner



Vulnerabilities

Vulnerabilities

Undefined Behavior

Example: CVE-2022-26446

BBC Home News Sport Reel Worklife Travel Future

NEWS

Home | War in Ukraine | Climate | Video | World | UK | Business | Tech | Science | Entertainm | More

World | Africa | Asia | Australia | Europe | Latin America | Middle East | US & Canada

Hawaii missile false alarm triggers shock, blame and apologies

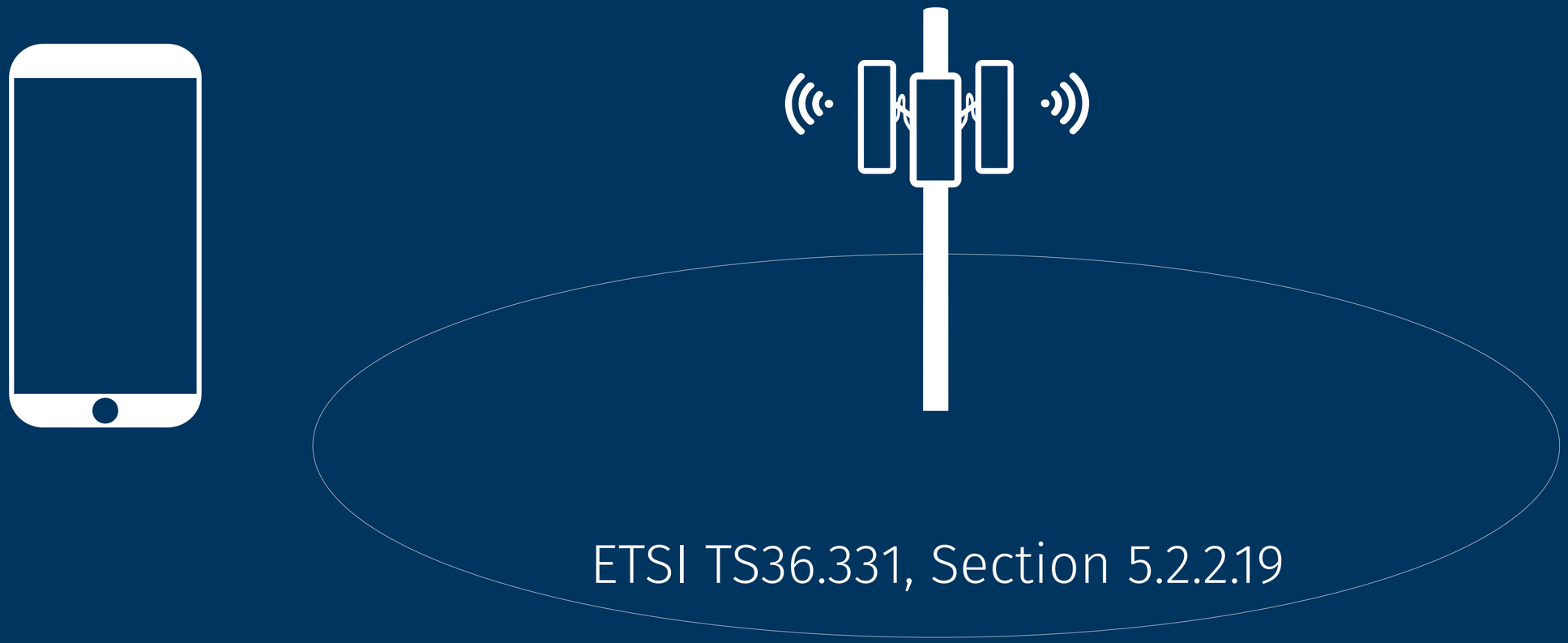
14 January 2018

People were warned to take shelter

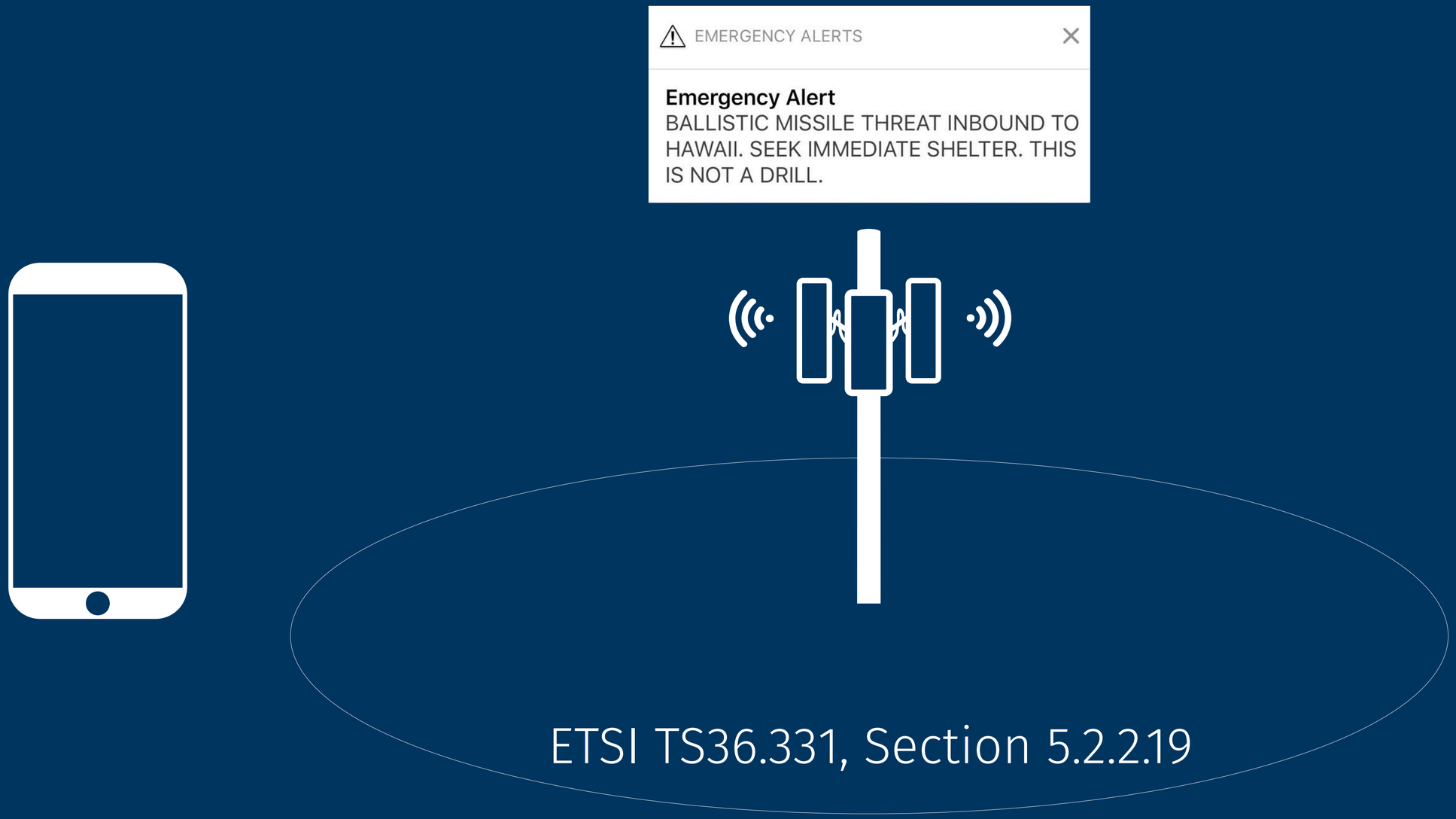
Residents and visitors in Hawaii have been recalling the shock of a false missile alarm, with many saying they thought they were going to die.

The alert of an incoming ballistic missile was sent wrongly on Saturday morning by an emergency system worker.

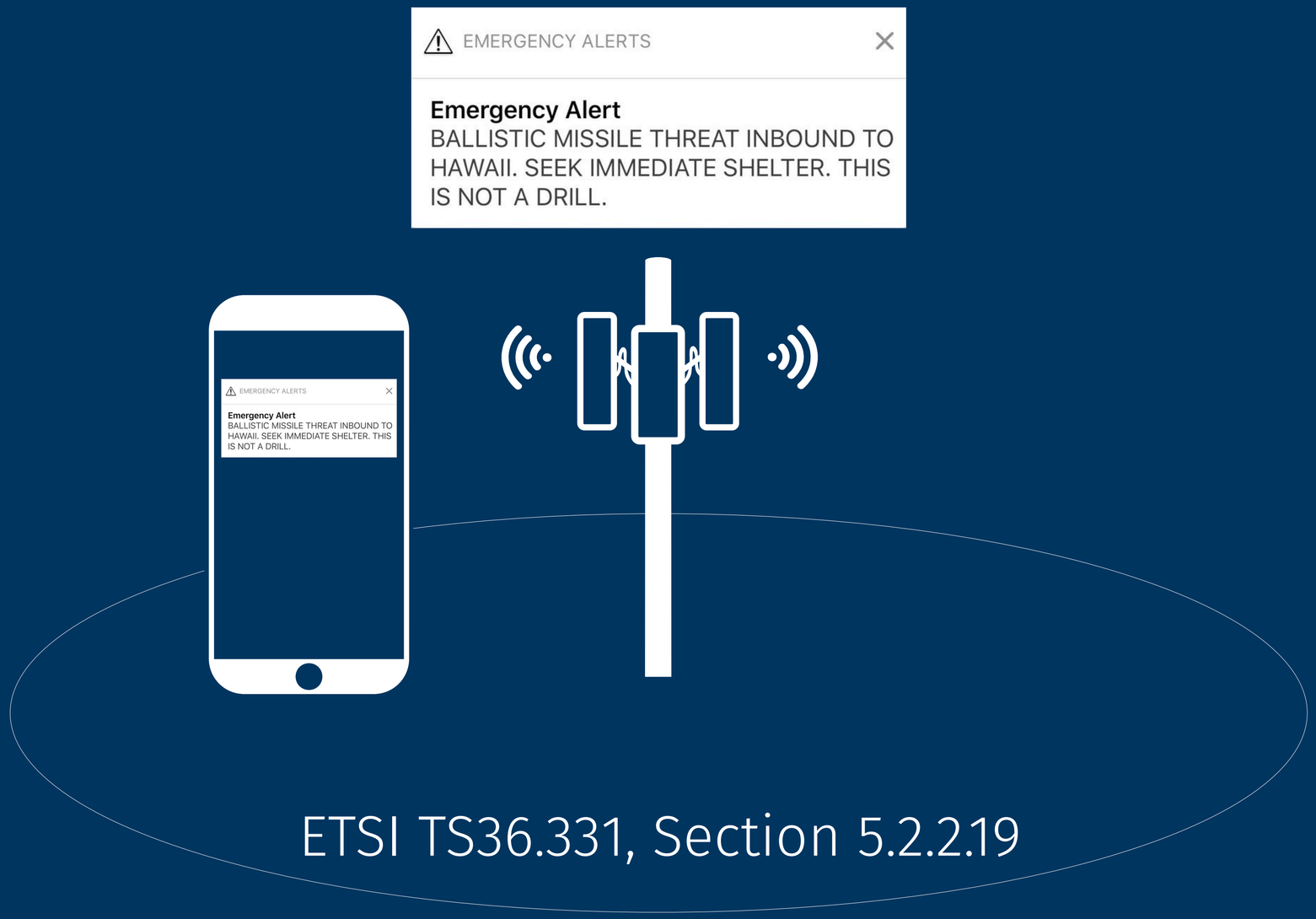
How your phone receives emergency alerts



How your phone receives emergency alerts



How your phone receives emergency alerts



ETSI TS36.331, Section 5.2.2.19

Behavior of MediaTek’s PWS implementation

Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND
Segment #: 2
lastSegment: y

Behavior of MediaTek’s PWS implementation

Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND
Segment #: 2
lastSegment: y

Behavior of MediaTek's PWS implementation

Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND
Segment #: 2
lastSegment: y

Behavior of MediaTek's PWS implementation

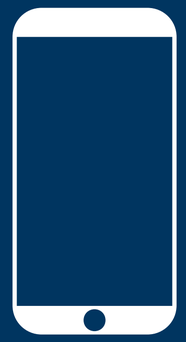
Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND
Segment #: 2
lastSegment: y



Baseband memory

Received segments: 0

Target segments: ?

Behavior of MediaTek's PWS implementation

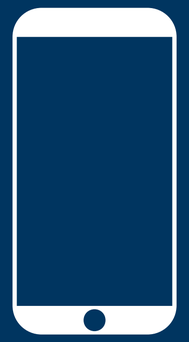
Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes



Baseband memory

INBOUND TO HAWAII

Received segments: 1 Target segments: ?

Behavior of MediaTek's PWS implementation

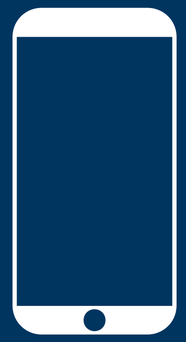
Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND TO HAWAI
Segment #: 2
lastSegment: yes



Baseband memory

INBOUND TO HAWAII

Received segments: 1

Target segments: 2

Behavior of MediaTek's PWS implementation

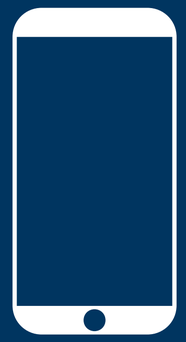
Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE T
Segment #: 1
lastSegment: no



Baseband memory

BALLISTIC MISSILE THREAT

INBOUND TO HAWAII

Received segments: 2

Target segments: 2

Behavior of MediaTek's PWS implementation

Text:

BALLISTIC MISSILE THREAT | INBOUND TO HAWAII

Text: BALLISTIC MISSILE THREAT
Segment #: 1
lastSegment: no

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE T
Segment #: 1
lastSegment: no

Baseband memory

BALLISTIC MISSILE THREAT INBOUND TO HAWAII

Received segments: 2 Target segments: 2

Undefined behavior in MediaTek's PWS implementation

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 3
lastSegment: no

Text: INBOUND
Segment #: 2
lastSegment: y

Baseband memory

Received segments: 0

Target segments: ?

Undefined behavior in MediaTek's PWS implementation

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE THREAT
Segment #: 3
lastSegment: no

Text: INBOUND TO HAWAI
Segment #: 2
lastSegment: yes

Baseband memory

INBOUND TO HAWAII

Received segments: 1 Target segments: 2

Undefined behavior in MediaTek's PWS implementation

Text: BALLISTIC MISSILE THREAT
Segment #: 3
lastSegment: no

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE T
Segment #: 3
lastSegment: no

Baseband memory

INBOUND TO HAWAII

BALLISTIC MISSILE THREAT

Received segments: 2

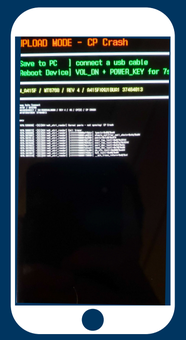
Target segments: 2

Undefined behavior in MediaTek's PWS implementation

Text: BALLISTIC MISSILE THREAT
Segment #: 3
lastSegment: no

Text: INBOUND TO HAWAII
Segment #: 2
lastSegment: yes

Text: BALLISTIC MISSILE T
Segment #: 3
lastSegment: no



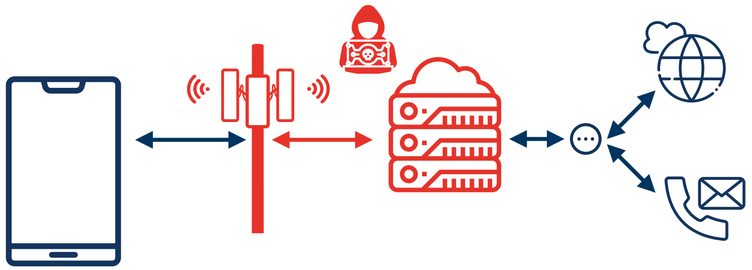
Baseband memory

<Uninitialized memory content> INBOUND TO HAWAII

BALLISTIC MISSILE THREAT

Received segments: 2 Target segments: 2

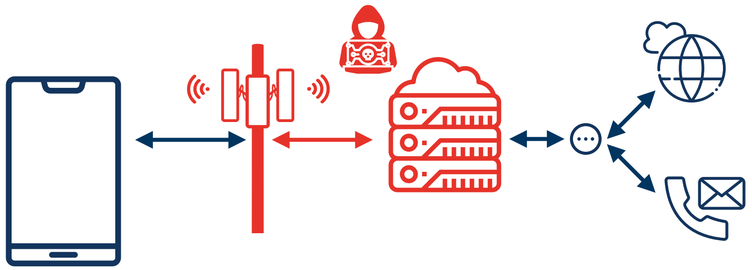
Attacker models - DoS via PWS



Malicious Mobile Network

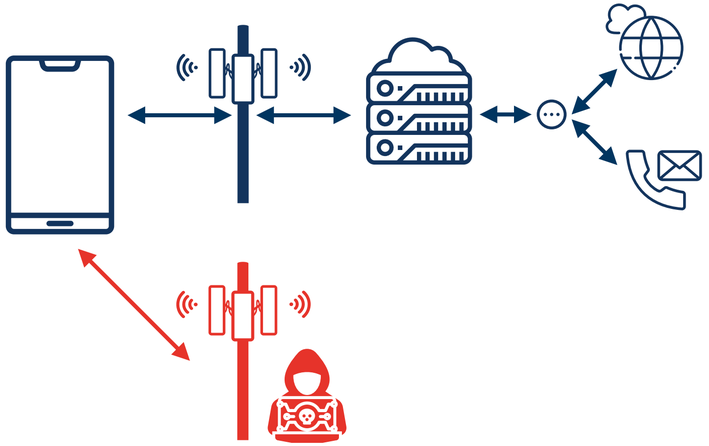
Has easier ways to perform a denial of service
→ **Theoretical threat**

Attacker models - DoS via PWS



Malicious Mobile Network

Has easier ways to perform a denial of service
→ **Theoretical threat**



Fake Base Station

Allows malicious actor to disable cellular communication
→ **No more emergency calls, data connectivity**

How did we discover this? Specifications.

ETSI TS36.331, Section 5.2.2.19

5.2.2.19 Actions upon reception of *SystemInformationBlockType12*

Upon receiving *SystemInformationBlockType12*, the UE shall:

1> if the *SystemInformationBlockType12* contains a complete warning message:

2> forward the received warning message, *messageIdentifier*, *serialNumber* and *dataCodingScheme* to upper layers;

2> continue reception of *SystemInformationBlockType12*;

1> else:

2> if the received values of *messageIdentifier* and *serialNumber* are the same (each value is the same) as a pair for which a warning message is currently being assembled:

3> store the received *warningMessageSegment*;

3> if all segments of a warning message have been received:

4> assemble the warning message from the received *warningMessageSegment*;

4> forward the received warning message, *messageIdentifier*, *serialNumber* and *dataCodingScheme* to upper layers;

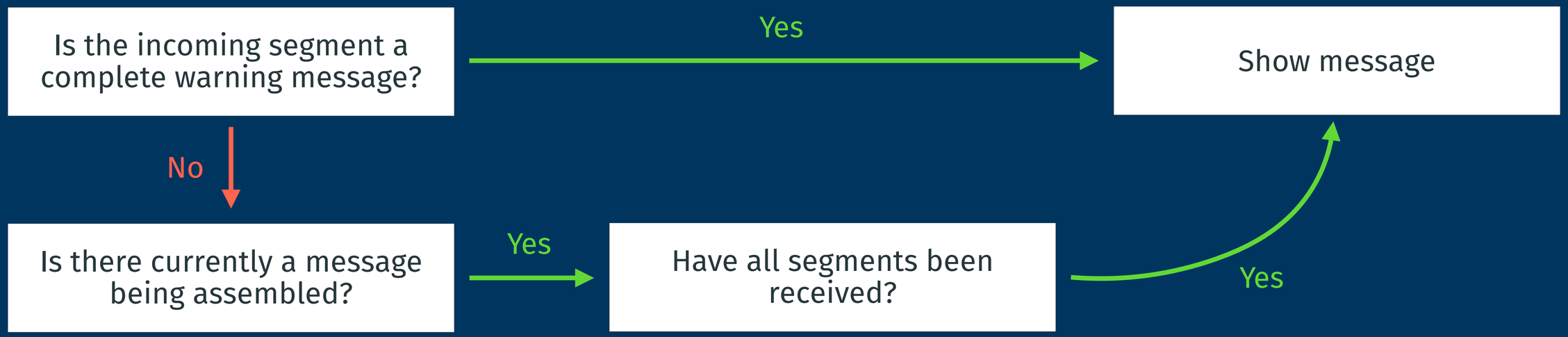
4> stop assembling a warning message for this *messageIdentifier* and *serialNumber* and delete all stored information held for it:

How did we discover this? Specifications.

ETSI TS36.331, Section 5.2.2.19

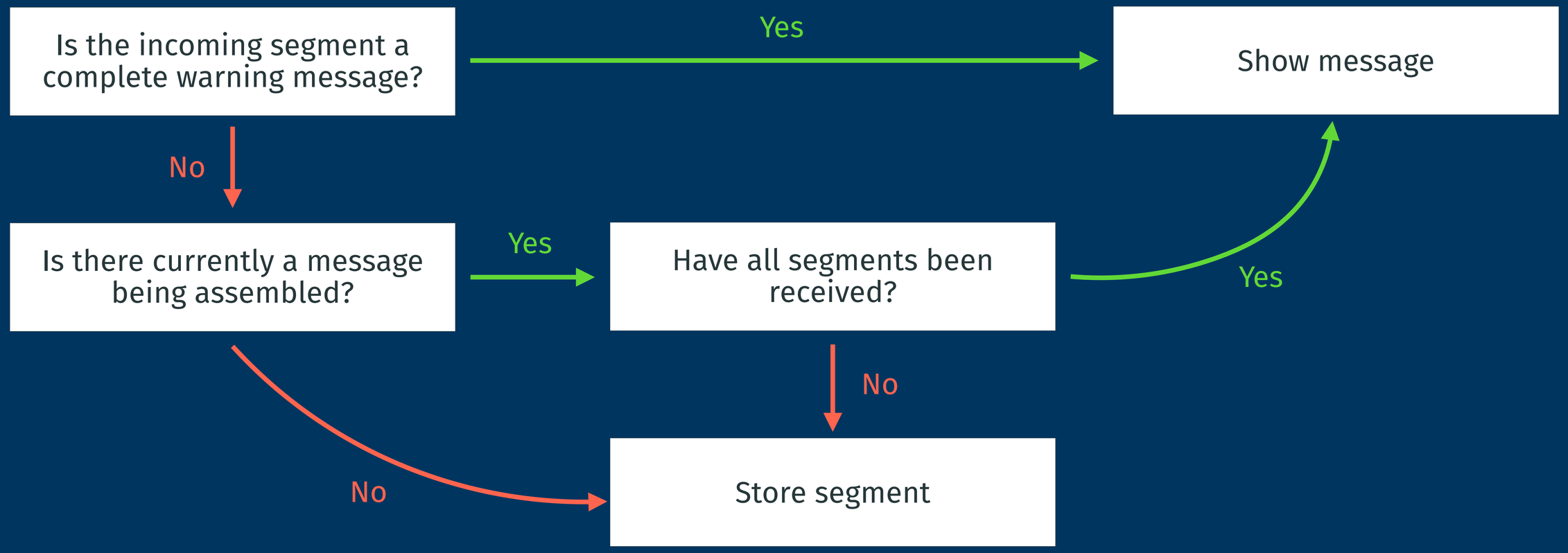
How did we discover this? Specifications.

ETSI TS36.331, Section 5.2.2.19



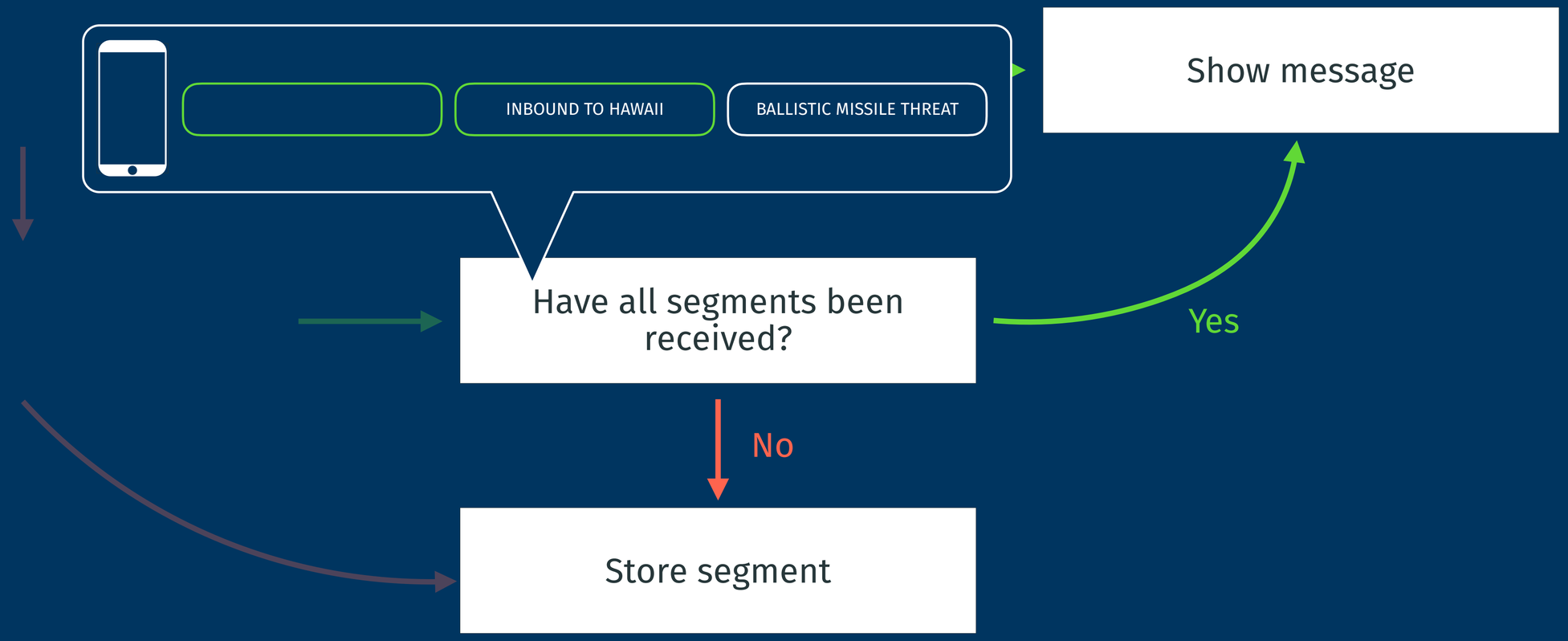
How did we discover this? Specifications.

ETSI TS36.331, Section 5.2.2.19



How did we discover this? Specifications.

ETSI TS36.331, Section 5.2.2.19



Vulnerabilities

Custom PDU formats

Example: CVE-2022-32591

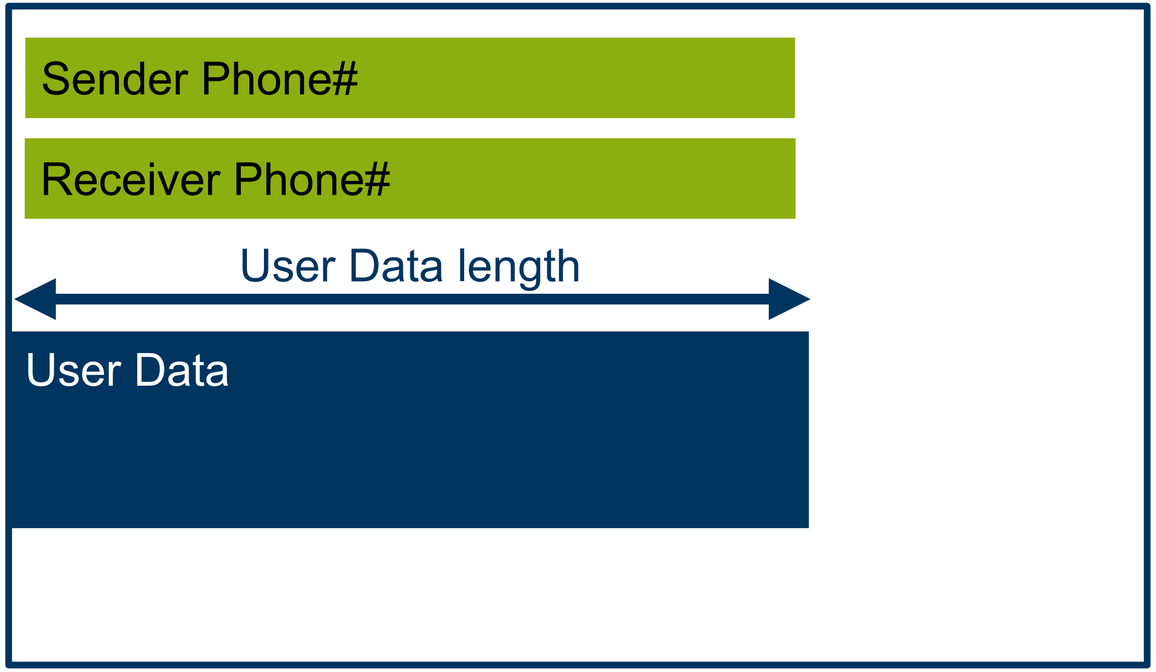
SMS

Since GSM, SMS is more than just text

- Optional functionality:
  - SMS reassembly (160 char limit!)
  - Various character sets
  - SIM card commands, sms-to-fax, ringtones

  → **User Data**

SMS PDU



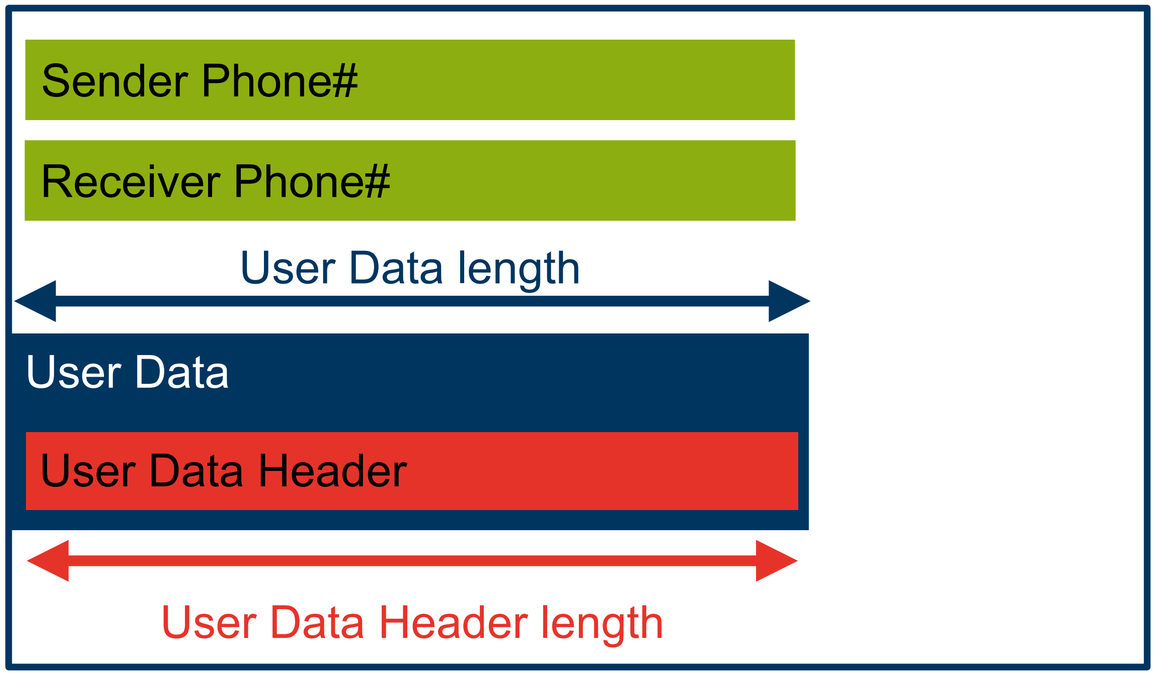
SMS

Since GSM, SMS is more than just text

- Optional functionality:
  - SMS reassembly (160 char limit!)
  - Various character sets
  - SIM card commands, sms-to-fax, ringtones

  → **User Data**
- Important for the vulnerability:
  - User data length field
  - User data header
    - Separate length field

SMS PDU



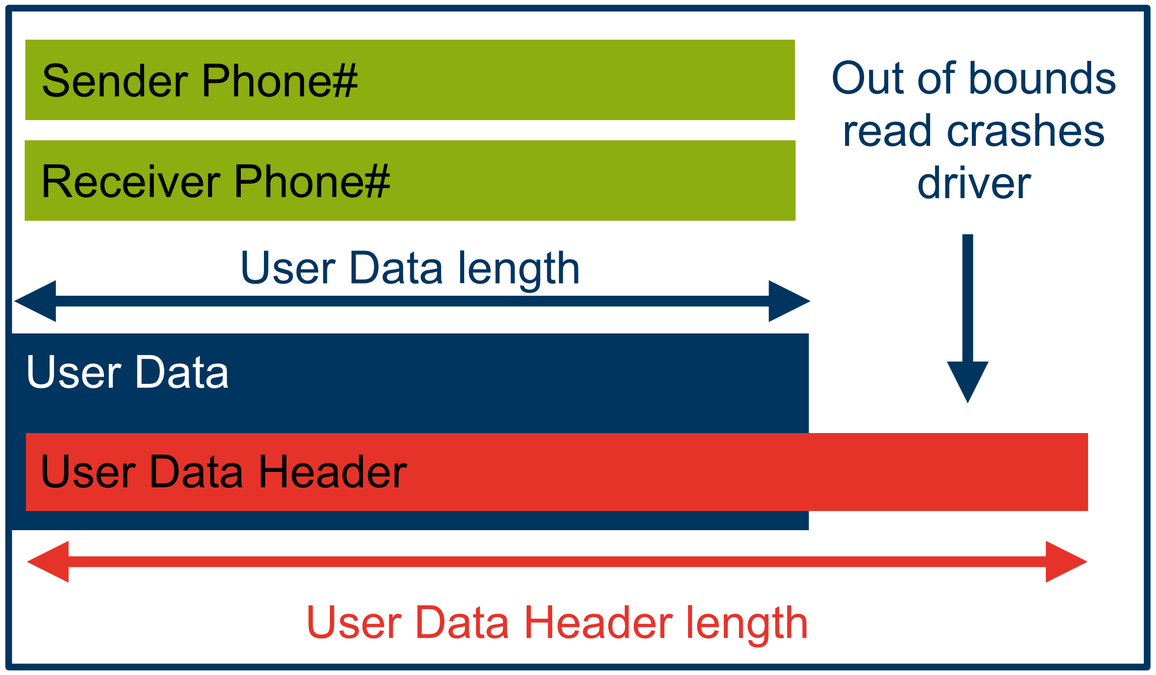
SMS

Since GSM, SMS is more than just text

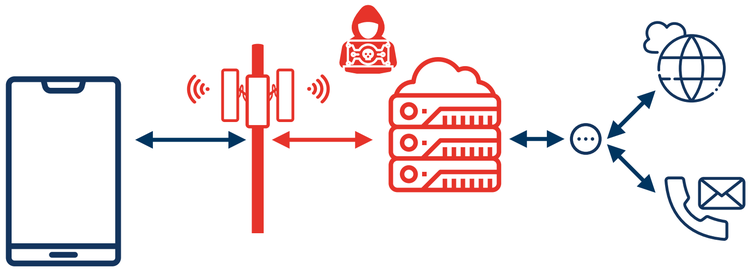
- Optional functionality:
  - SMS reassembly (160 char limit!)
  - Various character sets
  - SIM card commands, sms-to-fax, ringtones

  → **User Data**
- Important for the vulnerability:
  - User data length field
  - User data header
    - Separate length field

SMS PDU



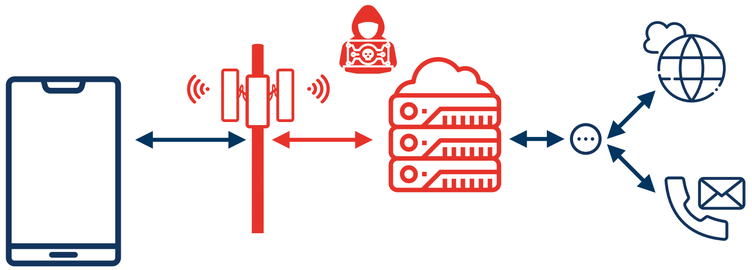
Attacker models - DoS via SMS



Malicious Mobile Network

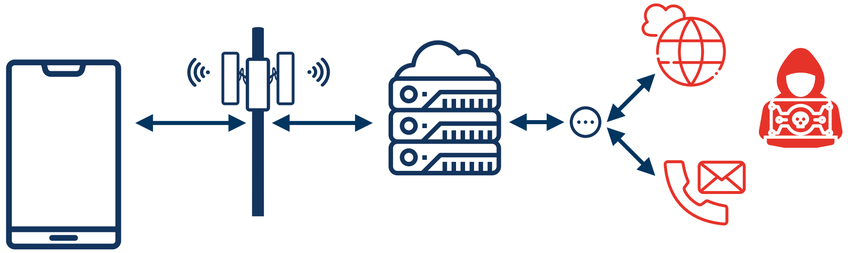
Has easier ways to perform a denial of service
→ **Theoretical Threat**

Attacker models - DoS via SMS



Malicious Mobile Network

Has easier ways to perform a denial of service
→ **Theoretical Threat**



Malicious Communication Partner

Allows malicious actor to disable cellular communication
→ **Disable affected phone from anywhere, just need to know phone number**

Underlying issues

Many protocol parts in LTE/5G use ASN.1:

Specification contains ASN.1 definitions

Baseband developers auto-generate parsers in C

→ **Limited attack surface**

Underlying issues

Many protocol parts in LTE/5G use ASN.1:

Specification contains ASN.1 definitions

Baseband developers auto-generate parsers in C

→ **Limited attack surface**

SMS uses a custom packet format instead

This has been inherited from GSM into LTE

→ **Simplifies backwards compatibility**

Underlying issues

Many protocol parts in LTE/5G use ASN.1:

Specification contains ASN.1 definitions

Baseband developers auto-generate parsers in C

→ **Limited attack surface**

SMS uses a custom packet format instead

This has been inherited from GSM into LTE

→ **Simplifies backwards compatibility**

→ **Requires manual implementation of a parser, description in spec is incomplete**

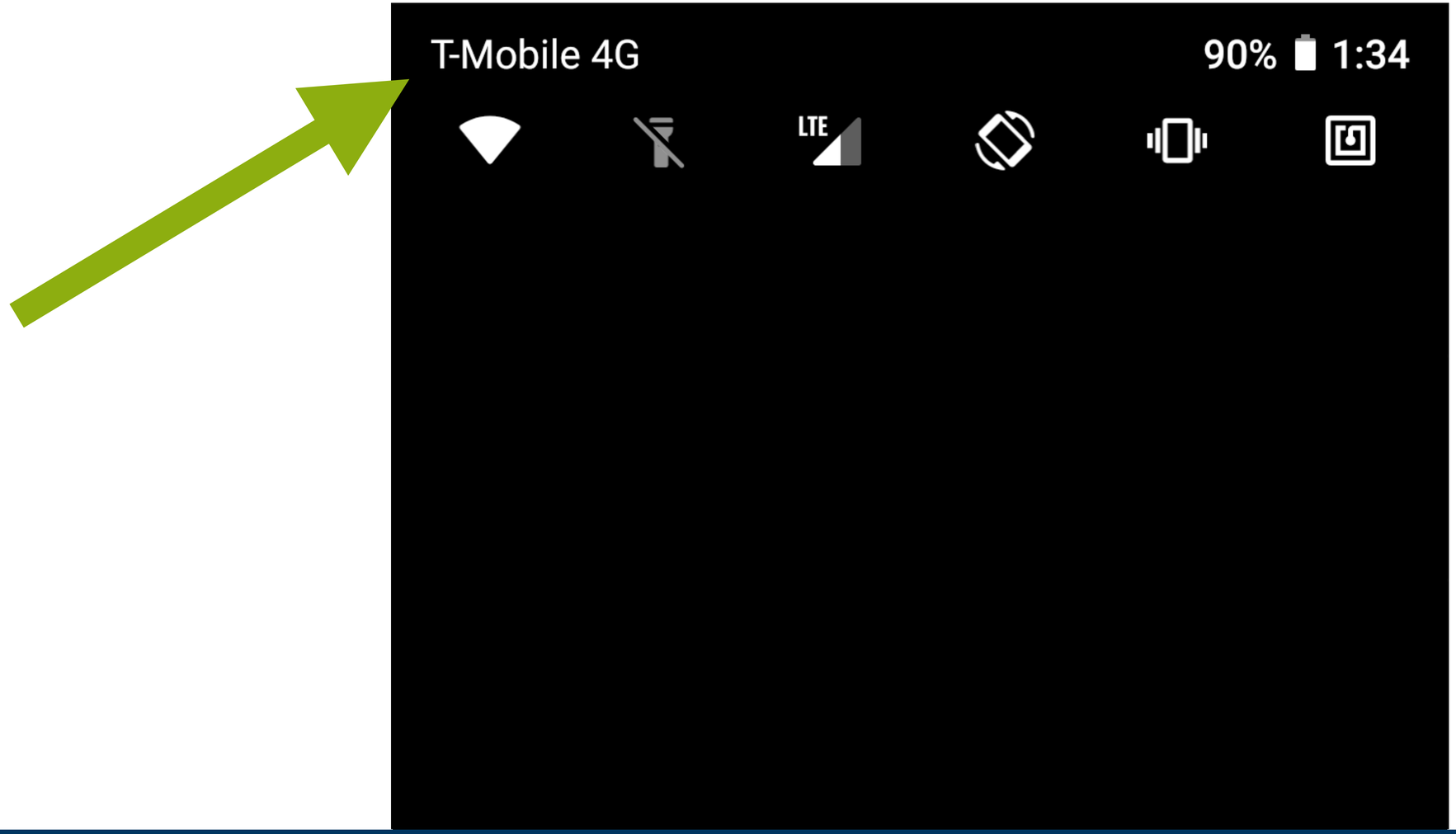
→ **Transitions issues from the 90s into today's standards**

Vulnerabilities

Custom field encodings

Example: CVE-2024-20039

Network operator names



Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

7 bit enc.: `1000001`

A

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

7 bit enc.: `10000011 000010`

A **B**

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

7 bit enc.: `10000011 00001010 00011`

A **B** **C**

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

7 bit enc.: `10000011 00001010 00011100 0100`

A B C D

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

7 bit enc.:	10000011	00001010	00011100	01001000	10110001	10100011	11001000
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

Android-side expects network name in modern ASCII (8-bit/character)

7 bit enc.:	10000011	00001010	00011100	01001000	10110001	10100011	11001000
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

Android-side expects network name in modern ASCII (8-bit/character)

7 bit enc.: `10000011 00001010 00011100 01001000 10110001 10100011 11001000`

A B C D E F G H

8 bit enc.: `01000001`

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

Android-side expects network name in modern ASCII (8-bit/character)

7 bit enc.: `10000011 00001010 00011100 01001000 10110001 10100011 11001000`

A	B	C	D	E	F	G	H

8 bit enc.: `01000001 01000010`

Decoding GSM 7-bit

Network names may be encoded in 7 bit/character encoding

Basebands store data in bytes (8-bit)

Android-side expects network name in modern ASCII (8-bit/character)

7 bit enc.:	10000011	00001010	00011100	01001000	10110001	10100011	11001000
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
8 bit enc.:	01000001	01000010	01000011	01000100	01000101	01000110	01000111	01001000

1 additional byte
per 8 characters

Decoding GSM 7-bit (MediaTek MT6768)

Target buffer for decoded name

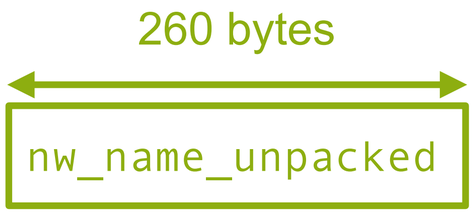
```
char[260] nw_name_unpacked;
if (type == GSM_7BIT) {
 csmss_gsm7_unpack(&nw_name_unpacked, &nw_name_packed, nw_name_len);
} else {
 memcpy(&nw_name_unpacked, nw_name_packed, nw_name_len);
}
```

Decoding GSM 7-bit (MediaTek MT6768)

Target buffer for decoded name

```
char[260] nw_name_unpacked;
if (type == GSM_7BIT) {
 csmss_gsm7_unpack(&nw_name_unpacked, &nw_name_packed, nw_name_len);
} else {
 memcpy(&nw_name_unpacked, nw_name_packed, nw_name_len);
}
```

Stack:



Decoding GSM 7-bit (MediaTek MT6768)

Target buffer for decoded name

7-bit encoded network name (max 255 bytes)

```
char[260] nw_name_unpacked;
if (type == GSM_7BIT) {
 csmss_gsm7_unpack(&nw_name_unpacked, &nw_name_packed, nw_name_len);
} else {
 memcpy(&nw_name_unpacked, nw_name_packed, nw_name_len);
}
```

Stack:

260 bytes

Decoding GSM 7-bit (MediaTek MT6768)

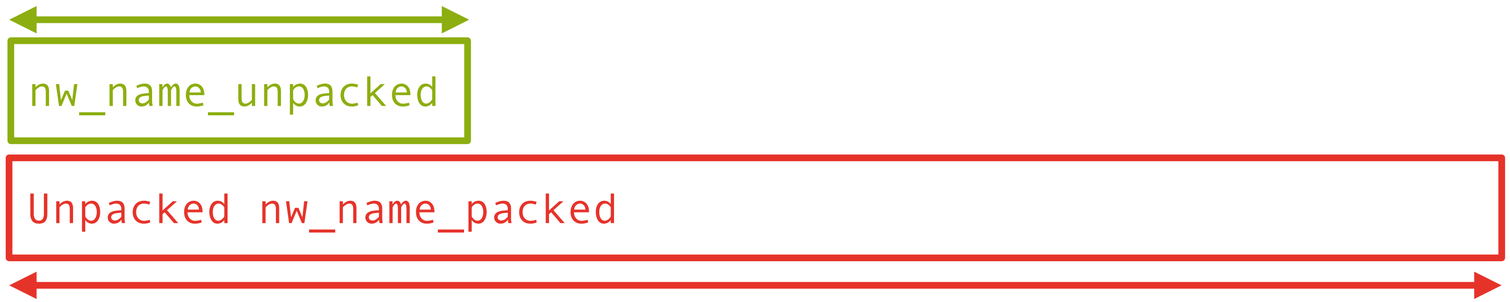
Target buffer for decoded name

7-bit encoded network name (max 255 bytes)

```
char[260] nw_name_unpacked;
if (type == GSM_7BIT) {
 csmss_gsm7_unpack(&nw_name_unpacked, &nw_name_packed, nw_name_len);
} else {
 memcpy(&nw_name_unpacked, nw_name_packed, nw_name_len);
}
```

Stack:

260 bytes



(255 * 8/7) = 292 bytes

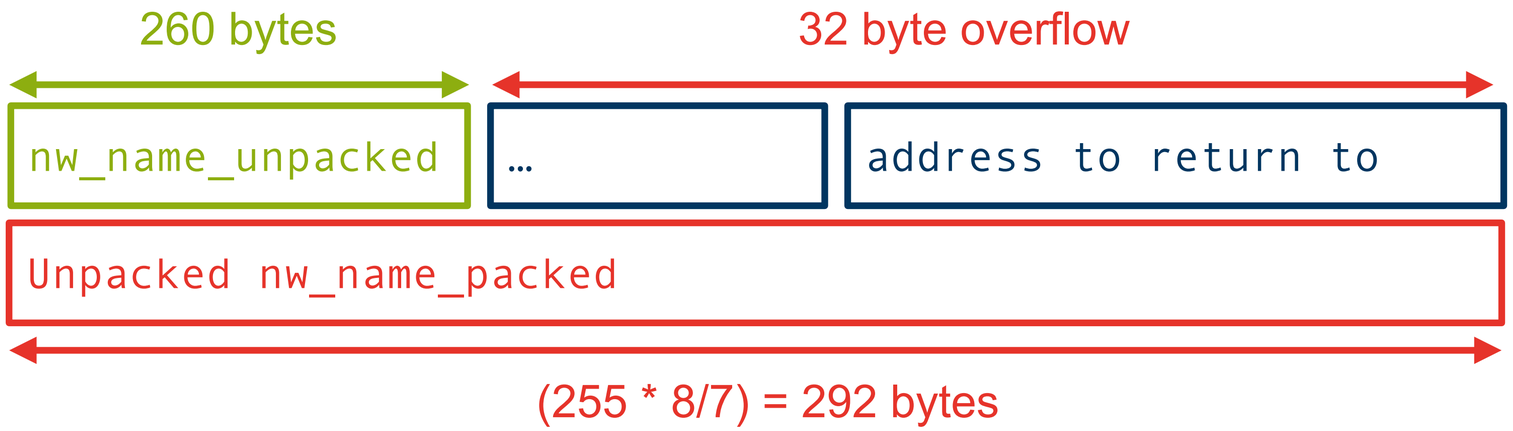
Decoding GSM 7-bit (MediaTek MT6768)

Target buffer for decoded name

7-bit encoded network name (max 255 bytes)

```
char[260] nw_name_unpacked;
if (type == GSM_7BIT) {
 csmss_gsm7_unpack(&nw_name_unpacked, &nw_name_packed, nw_name_len);
} else {
 memcpy(&nw_name_unpacked, nw_name_packed, nw_name_len);
}
```

Stack:



Underlying issue

This is just a silly implementation mistake, isn’t it?

Underlying issue

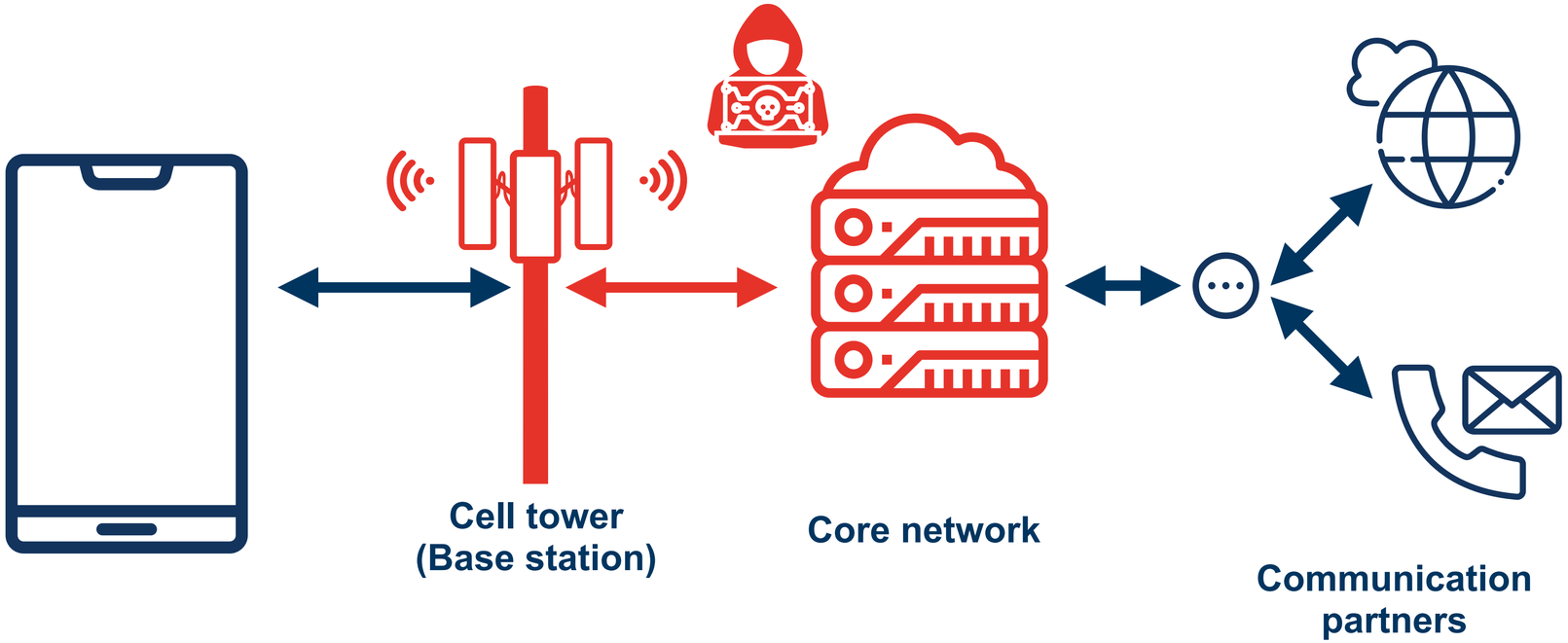
This is just a silly implementation mistake, isn't it?

Yes, but one that would not exist if the LTE spec wouldn't allow teletype era 7-bit encoding

→ **Simplifies backwards compatibility**

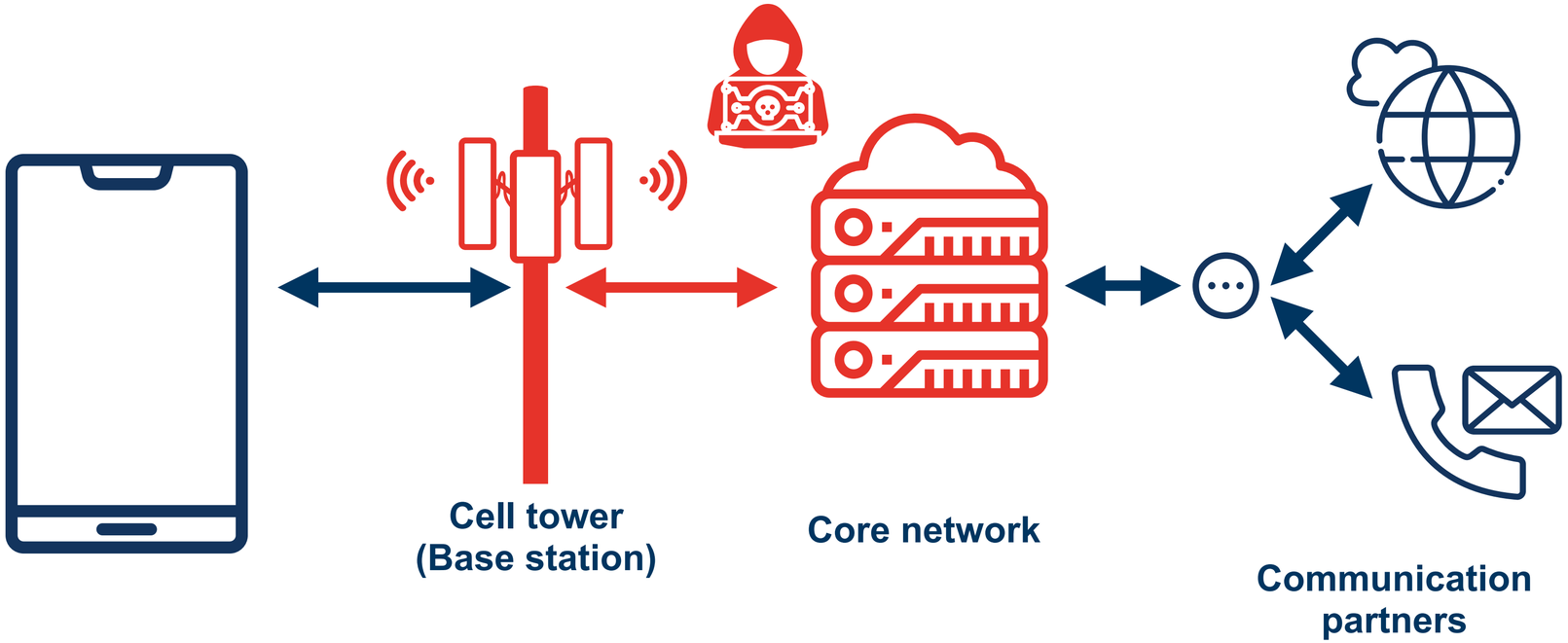
→ **Transitions issues from the ~~90s~~ 60s into today's standards**

Attacker model - RCE via Network Name



Malicious Mobile Network

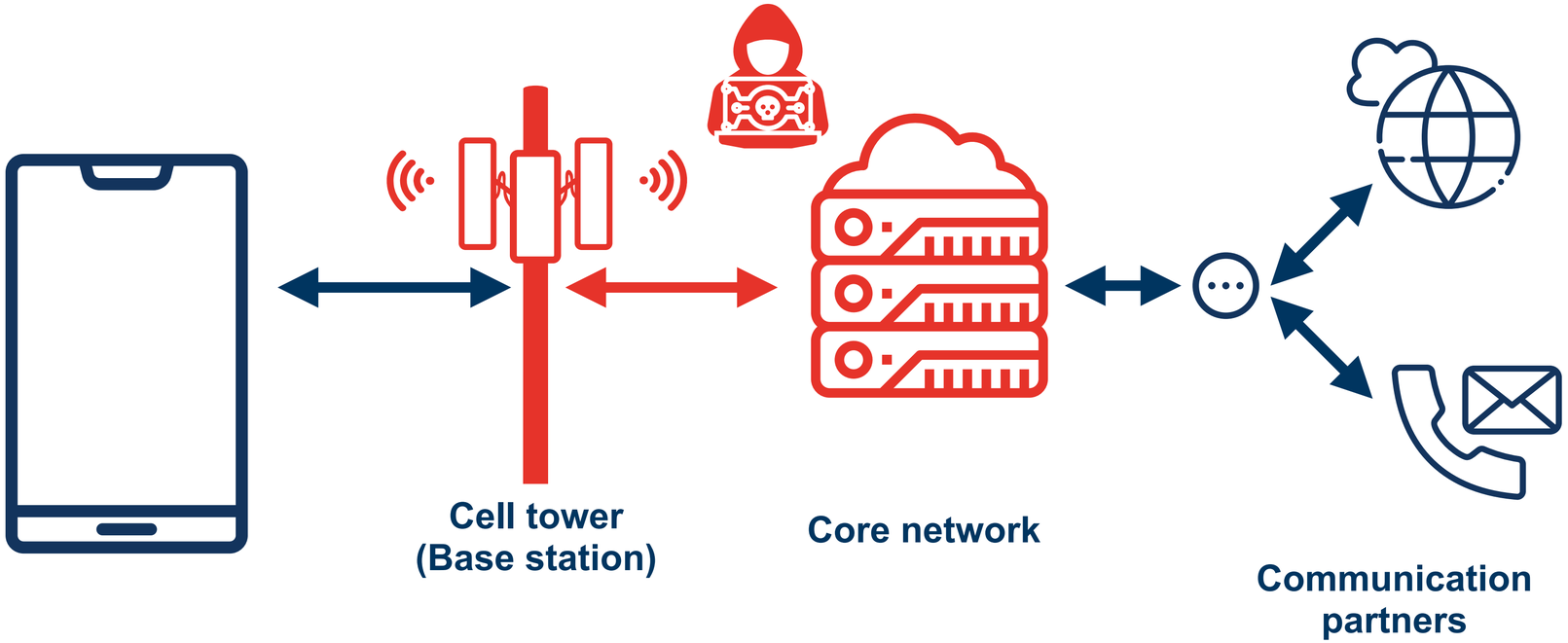
Attacker model - RCE via Network Name



Malicious Mobile Network

→ No ASLR, so only need to encode target program counter in 7 bit

Attacker model - RCE via Network Name



Malicious Mobile Network

→ No ASLR, so only need to encode target program counter in 7 bit

→ Potential lateral escalation to application processor

Attacker model

MEDIATEK

January 2025

Details

CVE	CVE-2024-20154
Title	Stack overflow in Modem
Severity	Critical
Vulnerability Type	RCE
CWE	CWE-121 Stack Overflow
Description	In Modem, there is a possible out of bounds write due to a missing bounds check. This could lead to remote code execution, if a UE has connected to a rogue base station controlled by the attacker, with no additional execution privileges needed. User interaction is not needed for exploitation.

Closing thoughts

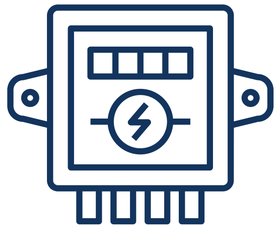
Beyond Smartphones

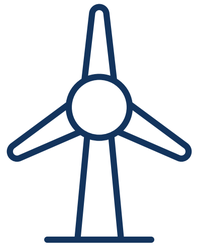
Cellular basebands are everywhere

Energy meters, wind turbine monitoring, eCall/V2X, GSM-R

→ **Impact of compromise depends on application**

→ **Update situation unclear**





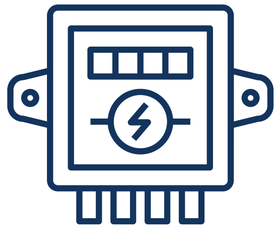
Beyond Smartphones

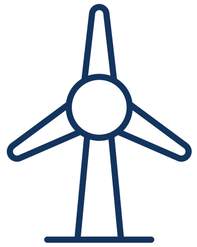
Cellular basebands are everywhere

Energy meters, wind turbine monitoring, eCall/V2X, GSM-R

→ **Impact of compromise depends on application**

→ **Update situation unclear**





Potential “solutions”:

1. Never assume cellular connectivity is always available
2. Always assume that the baseband may be compromised
3. Ensure the baseband’s firmware is updated regularly (monthly)

Takeaways:

Basebands are a viable attack vector, with sometimes trivial exploitability

Mitigations are often not up to par with those in Android/iOS

No easy & fast way to address these issues

Reach out:

daniel@danielklischies.net - https://www.danielklischies.net

https://informatik.rub.de/ubisys/