SPECIFICATION

Calypso Terminal API

Definitions

Version 1.0.0-SNAPSHOT, 2026-07-20

⚠ ***Warning. The present document cancels and replaces the previous release.***

Link and Contact

Website
https://calypsonet.org

Support
support@calypsonet.org

Table of Contents

1. Abstract

This document is the official technical specification of the **Terminal Definitions API** standardised by the **Calypso Networks Association** (CNA). It defines, in a language-agnostic way, the enumerated types that constitute **global constants** shared between several APIs of the CNA **Terminal API** family.

The Terminal Definitions API is a **foundation** module: it exposes no service interface, no SPI, no state machine. Its sole purpose is to host the cross-cutting enumerations that other Terminal APIs depend on.

Document Status

Reference	YYMMDD-SP-CNATerminalAPI-Definitions
Short name	CNA-TD-API
Version	1.0.0-SNAPSHOT
Revision date	2026-07-20
Editor	Calypso Networks Association
Source repository	https://github.com/calypsonet/calypsonet-terminal-definitions-uml-api
Reference license	Creative Commons Attribution-NoDerivatives 4.0 International (CC BY-ND 4.0)

The present document specifies the 1.0.0-SNAPSHOT of the Terminal Definitions API as an **initial release**. It defines a single, coherent baseline against which conforming implementations are evaluated.

Revision List

This section lists the high-level changes per version. The complete, fine-grained changelog is maintained in the `CHANGELOG.md` file at the root of the source repository.

Version / Date	Modifications
v1.0.0-SNAPSHOT 2026-07-20	Baseline.

2. Introduction

2.1. Purpose

The Terminal Definitions API centralises the **enumerated types** that constitute global constants shared between several CNA Terminal APIs. By isolating these notions in a dedicated module, the family avoids:

- duplicating the same enumerations across several APIs,
- coupling a consumer API to another consumer API merely because they share a constant,
- mixing the slow, conservative evolution rhythm of these constants with the faster rhythm of the consumer APIs.

2.2. Scope

This specification covers:

- the `RfTechnology` enumeration listing the radio-frequency communication technologies supported by contactless card readers,
- the `CardType` enumeration listing the card products and transport-protocol families identifiable at detection time.

The following topics are **out of scope**:

- the wire-level protocols associated with each technology,
- the discovery mechanism by which a reader determines the technology or type of a presented card,
- any interface or operation that consumes these enumerations—consumed APIs are responsible for documenting how they use them.

2.3. Conformance

A software product conforms to this specification if and only if:

1. it exposes the enumerations defined in Chapter 5 with the exact names and value sets prescribed here,
2. it preserves backward compatibility with respect to enumeration value identifiers across minor versions of the specification,
3. it preserves the structural relationships described in Chapter 4.

Throughout this specification, the key words **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY** and **OPTIONAL** are to be interpreted as described in RFC 2119 and RFC 8174 when, and only when, they appear in all capitals.

In conformance with RFC 2119, **SHALL** is equivalent to **MUST**; this specification uses **MUST** exclusively in order to avoid ambiguity.

2.4. Document Conventions

2.4.1. Naming

Type names follow `UpperCamelCase`; enumeration values use `UPPER_SNAKE_CASE`. Names defined by this

specification are reproduced as-is in the UML class diagram (Section 4.3).

2.4.2. Type tables

Each class and enumeration defined in this document is described by a table of the form:

Name	The type's normalized name.
Kind	The category of the type (class, enumeration, etc.).
Namespace	The namespace in which the type is defined.
Since	The version of the API in which the type was introduced.
Purpose	A normative description of the type's responsibility.
See also	Cross-references to related operations or types, each a hyperlink to the corresponding table. Present only when applicable.

The **Purpose** row states, in one or two sentences, the responsibility of the type and the way an instance is obtained. It is meant to be scanned, not read: any content that requires structure or depth—rules, lists, notes, value tables or examples—is placed as prose below the table.

2.5. References and Resources

2.5.1. Calypso References

This specification defines the shared vocabulary of the CNA Terminal API; it does not normatively reference any other CNA Terminal API document.

2.5.2. Normative References

Reference	Document
[RFC 2119]	*Key words for use in RFCs to Indicate Requirement Levels*, S. Bradner, March 1997.
[RFC 8174]	*Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, B. Leiba, May 2017.
[ISO/IEC 7816]	*Identification cards—Integrated circuit cards*.
[ISO/IEC 14443]	*Identification cards—Contactless integrated circuit cards—Proximity cards*.
[ISO/IEC 15693]	*Identification cards—Contactless integrated circuit cards—Vicinity cards*.

2.5.3. External Resources

Resource	Link
Calypso Networks Association	https://calypsonet.org
Terminal APIs website	https://terminal-api.calypsonet.org
Terminal APIs documentation	https://docs.terminal-api.calypsonet.org

3. Glossary and Acronyms

3.1. Glossary

Term	Definition
RF Technology	Radio-frequency communication technology used by a contactless reader to detect a card, as defined in ISO/IEC 14443.
Card Type	Identified product or transport-protocol family of a detected card.
Polling	The act, performed by a contactless reader, of broadcasting an interrogation pattern to discover a card in its field.

3.2. Acronyms

Abbreviation	Expansion
AID	Application IDentifier
CNA	Calypso Networks Association
ISO	International Organization for Standardization
RF	Radio Frequency
RFC	Request For Comments

4. Architectural Overview

4.1. Functional positioning

The Terminal Definitions API is depended upon by other CNA Terminal APIs that need to expose typed constants in their public signatures. The dependency is **public** (the enumerations appear in the consuming APIs' public surface) and is expected to be propagated transitively to the consumers of those APIs:

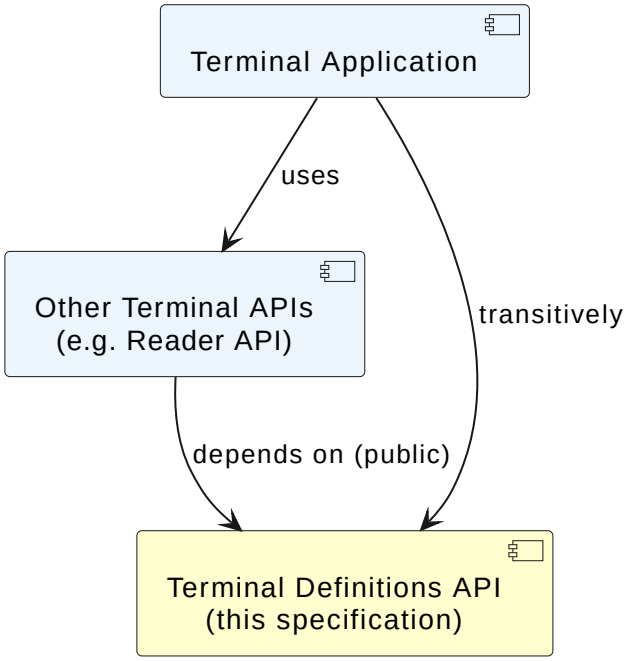

Figure 1. Terminal Definitions API — Architecture Overview

4.2. Logical namespaces

Namespace	Role	Summary
`definitions`	Public API	Shared vocabulary of the CNA Terminal API: the RF-technology and card-type enumerations (`RfTechnology`, `CardType`) and the API properties, reused across the terminal API specifications.

4.3. UML class diagram

The following UML class diagram provides a visual representation of the types and relationships defined by this specification:

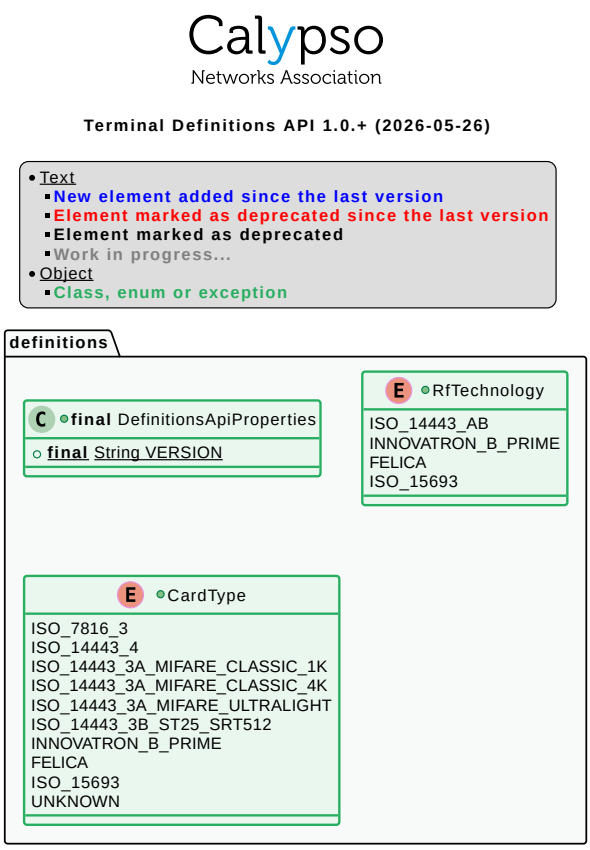

Figure 2. Terminal Definitions API — UML Class Diagram

4.4. Evolution policy

1. Adding a new value to an existing enumeration is a **backward-compatible** evolution and MAY occur in a minor version of this specification.
2. Removing or renaming a value is a **breaking** change and MUST only occur in a new major version of this specification.
3. Adding a new enumeration to the API is a **backward-compatible** evolution and MAY occur in a minor version.

5. API Specification

5.1. Classes

5.1.1. Definitions API Properties

Name	`DefinitionsApiProperties`
Kind	Final class
Namespace	`definitions`
Since	1.0
Purpose	Exposes the immutable properties of the Terminal Definitions API.

Constants

Name	Type	Since	Description
`VERSION`	`string`	1.0	String representation of the version of the API implemented by the conforming binding (e.g. `"1.0"`). The value is a dotted decimal of the form `MAJOR.MINOR`.

The `DefinitionsApiProperties` type cannot be instantiated; it exposes only static properties.

5.2. Enumerations

5.2.1. Card Type

Name	`CardType`
Kind	Enumeration
Namespace	`definitions`
Since	1.0
Purpose	Identifies a card type, as observed after detection. `CardType` is used as an **output** datum: it characterises what was actually detected, combining transport-protocol level and, when applicable, product identity.

Value	Since	Description
`ISO_7816_3`	1.0	Contact card compliant with ISO/IEC 7816-3.
`ISO_14443_4`	1.0	Contactless card compliant with the ISO/IEC 14443-4 transport protocol. No A/B distinction is exposed at this level: the transport layer is identical and the application behaviour does not depend on the underlying sub-layer.
`ISO_14443_3A_MIFARE_CLASSIC_1K`	1.0	MIFARE Classic 1 KB product (ISO/IEC 14443-3 Type A sub-layer).
`ISO_14443_3A_MIFARE_CLASSIC_4K`	1.0	MIFARE Classic 4 KB product (ISO/IEC 14443-3 Type A sub-layer).
`ISO_14443_3A_MIFARE_ULTRALIGHT`	1.0	MIFARE Ultralight product (ISO/IEC 14443-3 Type A sub-layer).
`ISO_14443_3B_ST25_SRT512`	1.0	ST25 SRT512 product (ISO/IEC 14443-3 Type B sub-layer).
`INNOVATRON_B_PRIME`	1.0	Innovatron B' card.

Value	Since	Description
`FELICA`	1.0	Sony FeliCa card.
`ISO_15693`	1.0	ISO/IEC 15693 vicinity card.
`UNKNOWN`	1.0	The card type could not be identified. May happen when a card of a type not yet referenced by this enumeration is detected, or when the detection process is partial.

Granularity rationale. The dual granularity of `CardType`—transport-protocol level for ISO/IEC 14443-4 vs product level for MIFARE / ST25—reflects the asymmetry between standardised transport (where the A/B sub-layer distinction does not affect application behaviour) and proprietary products (where the sub-layer is semantically tied to the product). This is why A/B is merged in `ISO_14443_4` but kept distinguishable in the product-specific values.

5.2.2. RF Technology

Name	`RfTechnology`
Kind	Enumeration
Namespace	`definitions`
Since	1.0
Purpose	Identifies a radio-frequency communication technology that a contactless reader can activate during polling. `RfTechnology` is used as an **input** parameter: the integrator declares, at detection time, which RF technologies the reader attempts to detect.

Value	Since	Description
`ISO_14443_AB`	1.0	ISO/IEC 14443 family, both Type A and Type B variants. The A/B distinction is **not** exposed at the polling level: the reader is asked to interrogate "the ISO 14443 family".
`INNOVATRON_B_PRIME`	1.0	Proprietary Innovatron B' technology.
`FELICA`	1.0	Sony FeliCa technology.
`ISO_15693`	1.0	ISO/IEC 15693 vicinity technology.