Introduction of Binary API at OTE markets

Jakub Šrom, Lukáš Vitásek

2. 6. 2026

Business area

- Introducing the interface change
- Timeline and trends

Technical area

- What is protobuf
- Data validation
- Changes in the AMQP interface
- Sequence of Broadcast Messages
- Digital signature

© OTE, a.s. 2026 – All rights reserved

Introducing the interface change

Changing the Automatic Communication Interface to the Binary API

- **Within the framework of the SIDC (Single Intraday Coupling)** project to ensure adequate **performance and stability of the central XBID system (change of the OTE-COM – XBID interface)**
- **The XML message format will be replaced by binary content, the Protocol Buffers data format.**
- The concept of communication (existing communication scenarios and the AMQP communication protocol) does not change with the transition to new communication scenarios.
- **To ensure the benefits of more efficient and faster communication for OTE market participants, it is necessary to adapt the automatic communication interface to the intended binary format Protocol Buffers also on the side of CS OTE (AMQP communication)**

© OTE, a.s. 2026 – All rights reserved

Information on the transition to the Binary API for automatic communication on the intraday electricity and gas market

Modifying AMQP automatic communication to:

- Intraday continuous electricity market
- Intraday gas market

→ **The XML message format will be replaced by the binary Protocol Buffers data format.**

The modification does not affect:

- users of the OTE-COM application for the intraday continuous electricity market
- users of the OTE-COM application for the intraday gas market
- Mobile applications
- CS Portal - short-term markets (DAM and IDA) and Registration of Realization Diagrams

© OTE, a.s. 2026 – All rights reserved

The change in the communication format will take place as part of the automatic communication of Intraday Continuous Electricity and Gas Market towards market participants

The documentation provided includes:

- **Updated version of the Electricity Message Formats Document**
- **Updated version of the gas message format manual**

XLSX mapping tables that mapping of the existing XML attribute names to their corresponding Binary API attribute names

- XLSX Mapping Tables Electricity
- XLSX Gas Mapping Table

Protobuf definition (.proto files):

- For electricity (IM)
- For Gas (IMP)

OTE Spojujeme trhy a příležitosti

Pro registrované účastníky a výrobce POZE: Portál CS OTE | Sandbox | Sandbox2 | EN

O společnosti | Smluvní partneři | Risk management | Krátkodobé trhy | Statistika | Dokumentace | Environmentální certifikáty | Podpora a výrobní zdroje | Pro média | Kariéra

Dokumentace

Informace k přechodu na rozhraní Binary API pro automatickou komunikaci na Vnitrodenním trhu s elektřinou a plynem

Informace k přechodu na rozhraní Binary API pro automatickou komunikaci na Vnitrodenním trhu s elektřinou a plynem

V rámci automatické komunikace AMQP na Vnitrodenním kontinuálním trhu s elektřinou a Vnitrodenním trhu s plynem se plánuje úprava rozhraní, kdy XML formát zpráv bude nahrazen binárním datovým formátem Protocol Buffers. Při používání aplikací Vnitrodenního trhu s elektřinou a plynem a mobilních aplikací nedochází k žádným změnám na straně uživatele.

Rozhraní automatické komunikace Vnitrodenního trhu kontinuálního s elektřinou (VDT) a Vnitrodenního trhu s plynem (VDP) vůči účastníkům trhu je v současnosti realizováno prostřednictvím AMQP protokolu s přenosem zpráv v XML formátu. Projekt SIDC (Single Intraday Coupling – Jednotný propojený vnitrodenní trh) bude z důvodu nutnosti zajištění odpovídajícího výkonu a stability centrálního systému XBID, i s ohledem na budoucí rozvoj vnitrodenního kontinuálního obchodování a legislativní požadavek pro přechod alokace vnitrodenních přeshraničních kapacit na mechanismus Flow-based, zavádět změnu komunikačního formátu zpráv pro datové výměny mezi centrálním řešením XBID a lokálními systémy na straně NEMOs (Nominated Electricity Market Operator – Nominovaný organizátor trhu s elektřinou). **XML formát zpráv bude nahrazen binárním obsahem, datovým formátem Protocol Buffers.** Koncept komunikace (stávající portfolio komunikačních scénářů a komunikační protokol AMQP) se z pohledu přechodu na nové komunikační scénáře na straně XBID nemění. Ve světle této chystané změny bude nutné provést adaptaci komunikačního rozhraní na zamýšlený binární formát Protocol Buffers i na straně CS OTE, aby mohl být nárůst výkonnosti a rychlosti komunikace přenesen i na rozhraní mezi OTE a účastníky trhu.

Ke změně komunikačního formátu dojde v rámci automatické komunikace VDT i VDP vůči účastníkům trhu. Poskytnutá dokumentace zahrnuje následující:

- **Aktualizovaná verze manuálu formátů zpráv elektřina a plyn** zohledňující nový komunikační formát zpráv, dále jsou k dispozici *XLSX mapovací tabulky elektřina a plyn*, které poskytují mapování stávajících názvů XML atributů na nové názvy BIN API ekvivalentů těchto atributů – viz materiály ke stažení níže.
- **Protobuf definice (.proto soubory):**
  - Za elektřinu (VDT) ke stažení níže
  - Za plyn (VDP) ke stažení níže

Webinář pro účastníky trhu k detailnímu představení změn souvisejících se zavedením Binary API je naplánován na úterý 2. 6. 2026 od 14:00 do 16:30. Pozvánky byly distribuovány účastníkům trhu prostřednictvím emailové komunikace.

Následně očekáváme:

- **Zahájení testování s ÚT v září/říjen 2026**
- **Uvedení změny do provozu v Q1 2027**

Bližší informace a detaily k plánovanému testování budou pravidelně poskytovány prostřednictvím webových stránek OTE a emailové komunikace.

POPIS	SOUBORY KE STAŽENÍ
Formáty zpráv elektřina	D1.4.X Formáty zpráv Binary API OTE-COM ELE_verzeC D1.4.X Formáty zpráv Binary API OTE-COM ELE_verzeC_revize (revize oproti předchozí verzi B)
XLSX mapovací tabulky elektřina, ze kterých lze vyčíst mapování stávajících názvů XML atributů na nové názvy BIN API ekvivalentů těchto atributů	Změna formátu zpráv OTE-COM ele protobuf vs XML_verzeB (zůstává kompatibilní s verzí C Formátů zpráv elektřina)
Protobuf definice (.proto soubor) elektřina	OTE-COM ele protobuf
Formáty zpráv plyn	D1.5.1 Formáty zpráv Binary API OTE-COM GAS_verzeB D1.5.1 Formáty zpráv Binary API OTE-COM GAS_verzeB_revize (revize oproti předchozí verzi A)
XLSX mapovací tabulky plyn ze kterých lze vyčíst mapování stávajících názvů XML atributů na nové názvy BIN API ekvivalentů těchto atributů	Změna formátu zpráv OTECOM gas protobuf vs XML_verzeB
Protobuf definice (.proto soubor) plyn	OTE-COM gas protobuf

Dokumentace k přístupu do systému OTE
XSD, WSDL, služby veřejného webu OTE
Dokumentace - Elektřina
Dokumentace Elektřina
Dokumentace Trhy
Užitečné informace pro PDS/PLDS
Informace k přechodu na 15 min. zúčtovací a obchodní periodu u komodity elektřina
Informace k přechodu na rozhraní Binary API pro automatickou komunikaci na Vnitrodenním trhu s elektřinou a plynem
Dokumentace TDD
Informace k zavedení flexibility a akumulace v CS OTE od 1.8.2026
Dokumentace - Plyn
Aktuality plyn
Užitečné informace
Úsporný tarif
Kompenzace cen elektřiny a plynu

→ You can download the documents at the dedicated page for the Binary API on the OTE website here

www.ote-cr.cz

© OTE, a.s. 2026 – All rights reserved

Timeline and trends

Timeline in Binary API implementation:

- Initial information on market participants (17.12.2025)
- Published Protobuf definitions (.proto files):
  - For electricity (VDT) (21.04.2026)
  - For gas (VDP) (25.05.2026)

- **Webinar on Binary API** (02.06.2026 at 14:00)
  - Send any questions to market@ote-cr.cz
  - The presentation will be published on the OTE website
  - We will continuously publish a document with any questions and answers

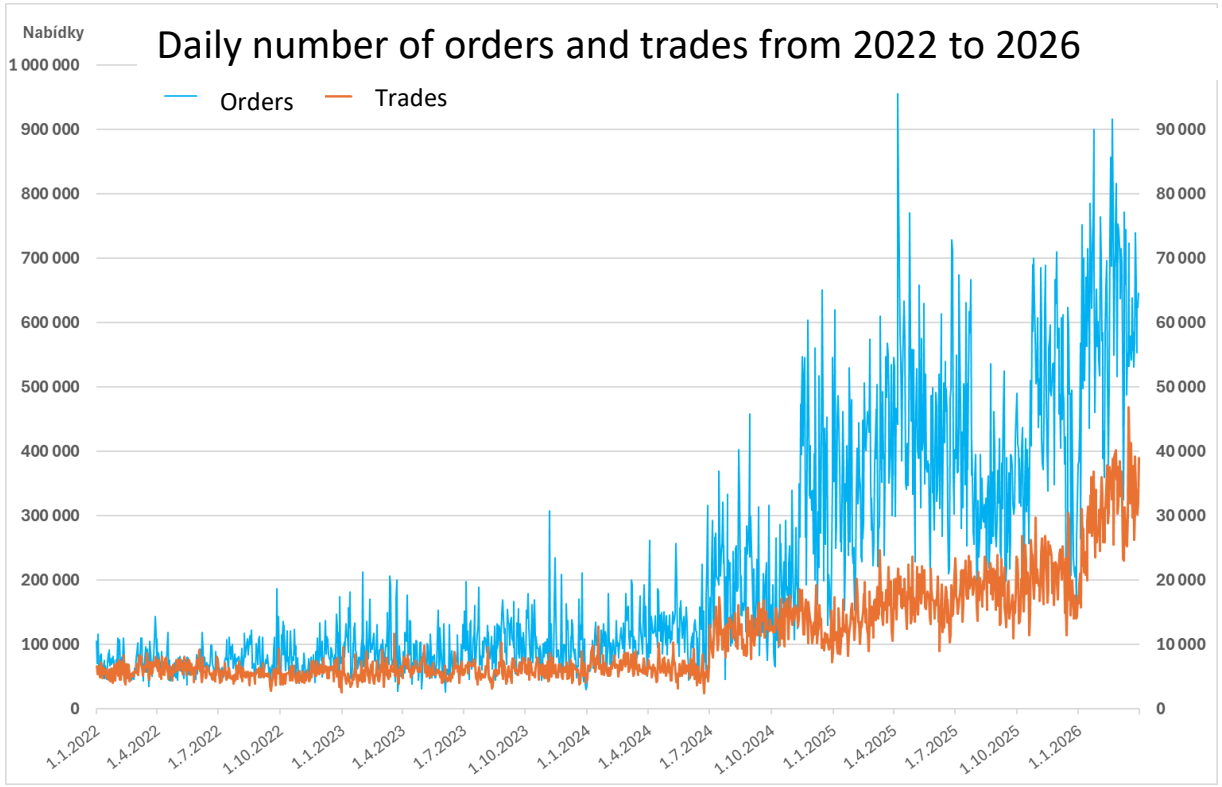
- **Start of testing with Market Participants (September/October 2026)**
  - You will be informed in advance about the details of the start of testing

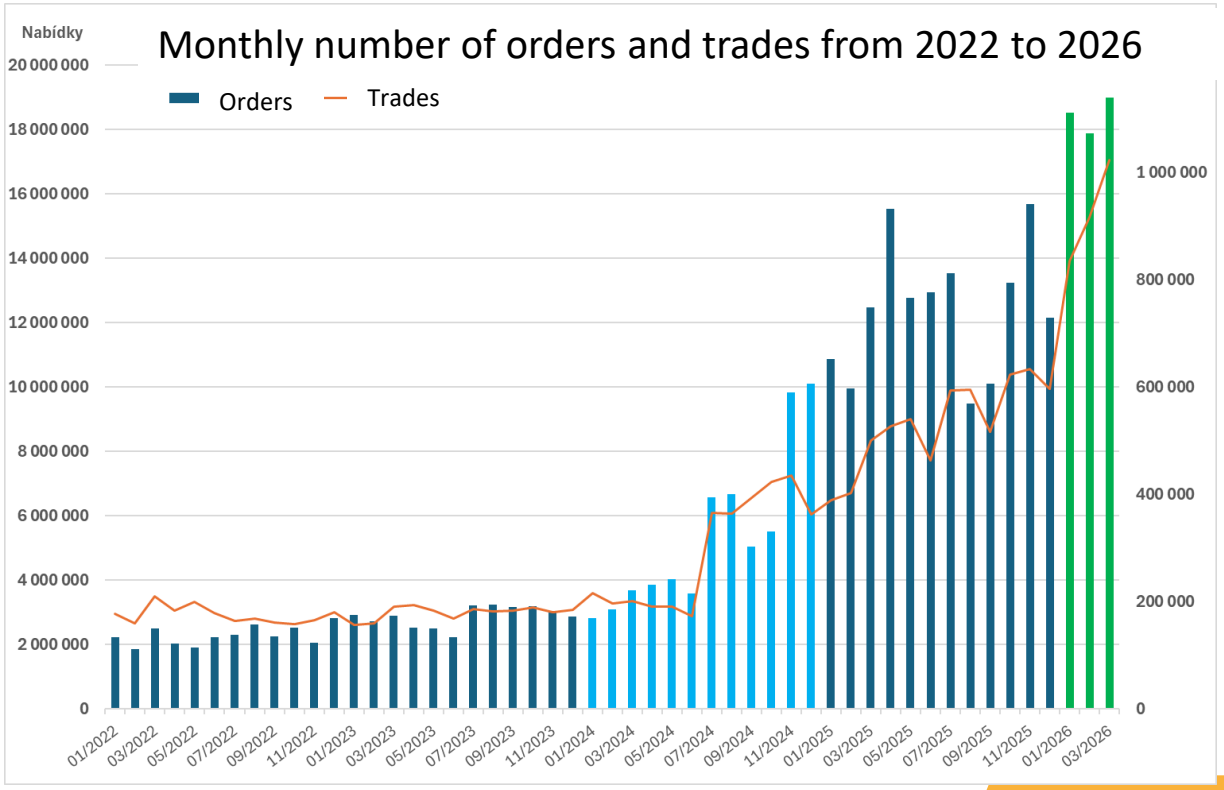
- **Launch of Binary API under SIDC and OTE (Q1 2027)**

© OTE, a.s. 2026 – All rights reserved

OTE-COM

- The Trend of Algorithmic Trading Across Europe – Engaging Robots
- Steep increase in orders vs trades
- More trading close to real-time

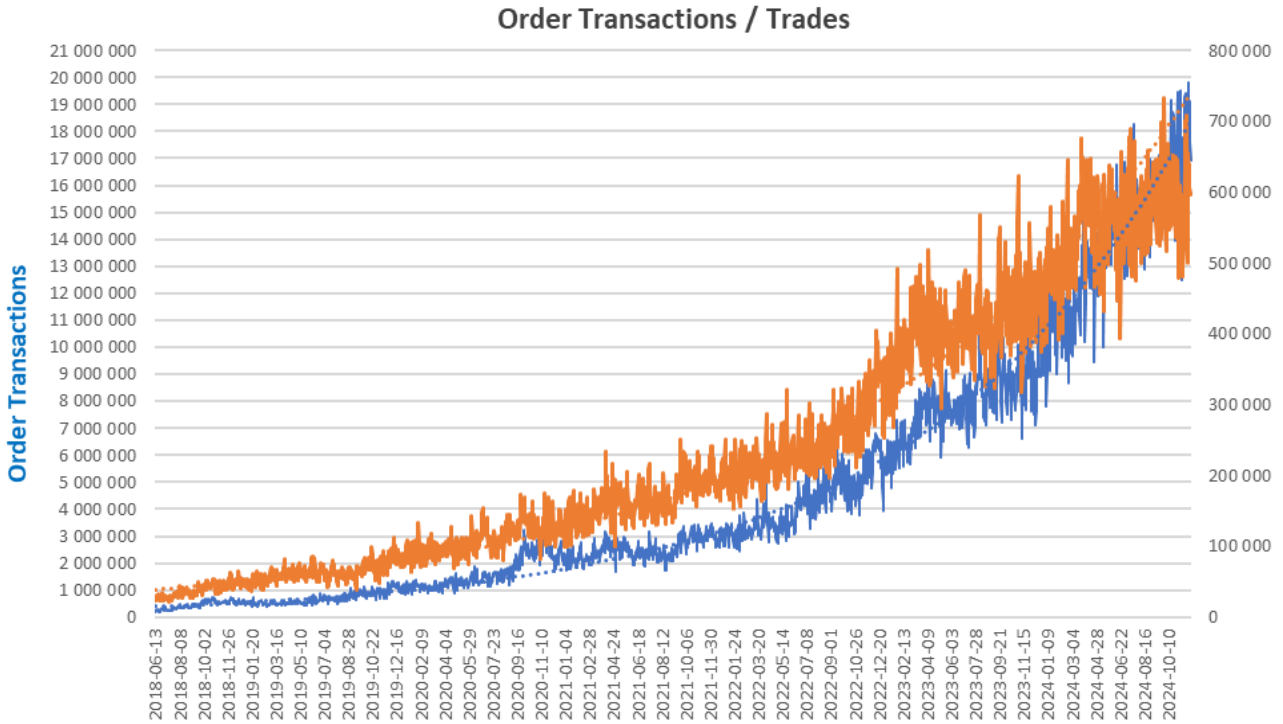


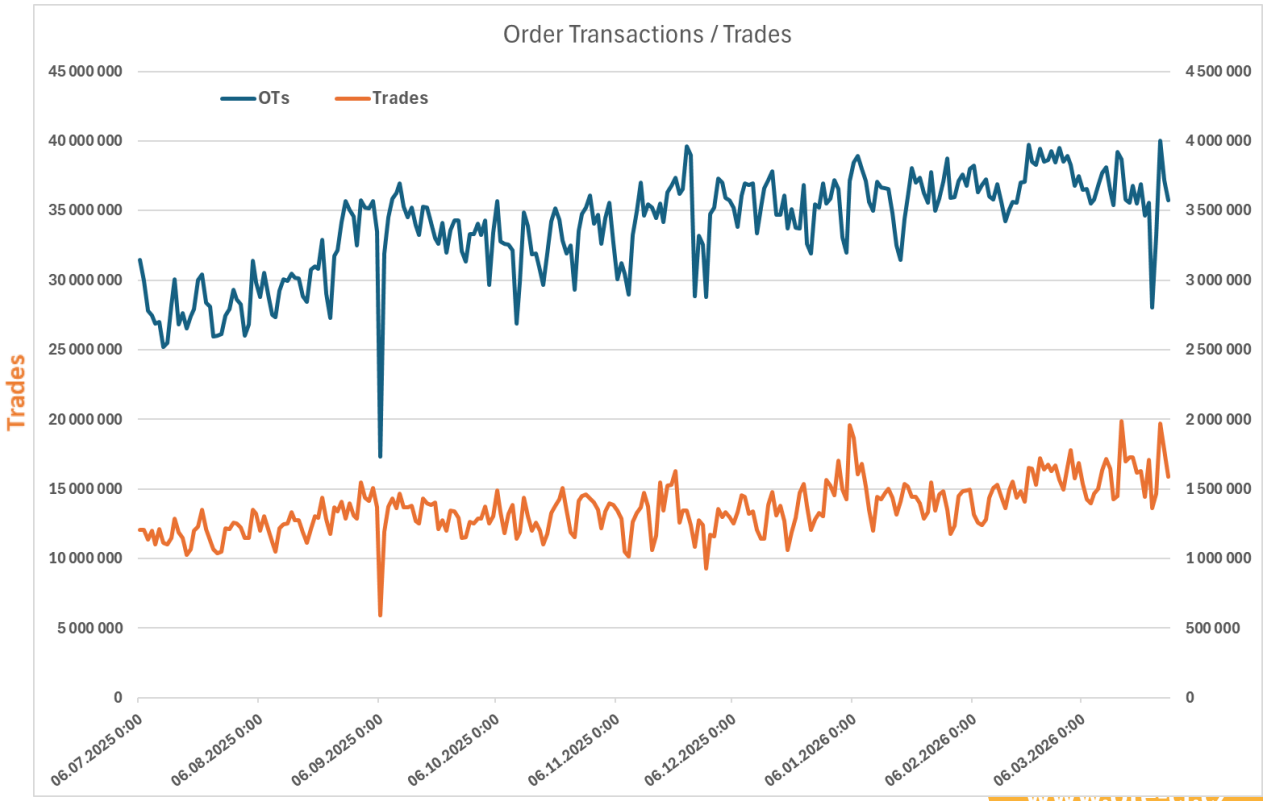


© OTE, a.s. 2026 – All rights reserved

Developments in SIDC

- Daily numbers of orders and trades within the SIDC since the first trading day (June 13, 2018)





© OTE, a.s. 2026 – All rights reserved

What is protobuf

What is protobuf

Why this change?

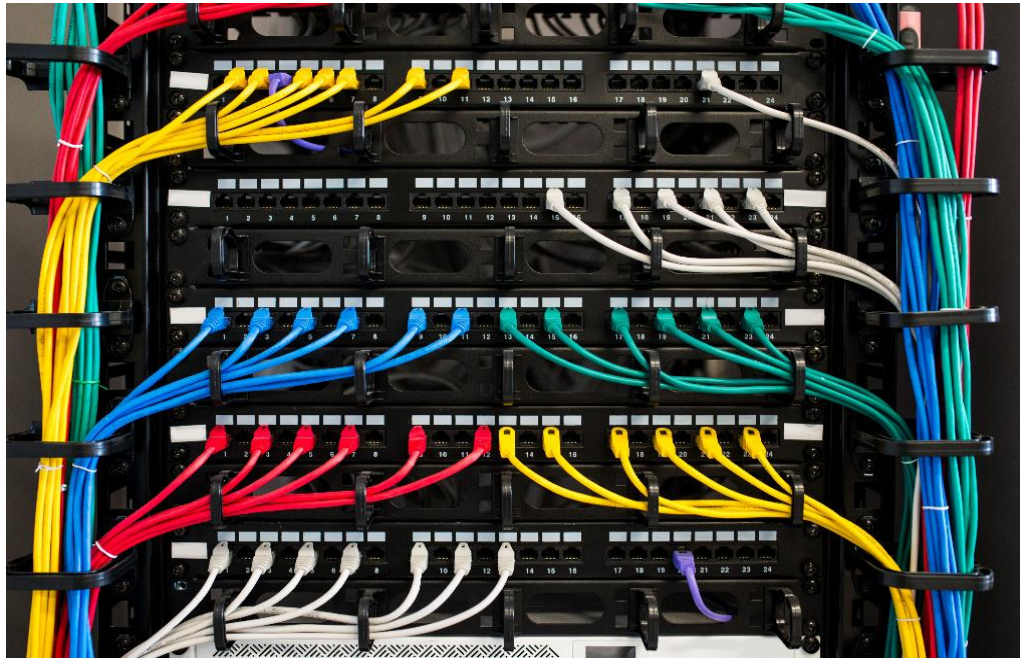
- Transaction load within SIDC has been growing significantly in recent years
  - This is driven by a general increase in trading activity
  - Additionally, the introduction of 15-minute products has led to a substantial increase in the number of contracts
    - As a result, the number of distributed updates (not only of the public order book) has increased proportionally

- The current message transmission approach is reaching its limits
  - XML (de)serialization and validation are time-consuming
  - XML itself is a relatively verbose data format
    - Messages are larger than strictly necessary
- XBID addresses this issue by migrating to the Protobuf format

© OTE, a.s. 2026 – All rights reserved

What is protobuf

- A data format developed by Google for serializing structured data into a binary representation
- Designed with an emphasis on simplicity and performance
- Compared to other formats such as JSON and XML, it offers messages that are 2–4× smaller
- Project website: https://protobuf.dev/



© OTE, a.s. 2026 – All rights reserved

What is protobuf

- Message structure is defined in a textual .proto schema
  - Similar to XSD, it defines messages and their structure
  - By default, it does not support validation rules
  - The schema is compiled using the protoc tool into source code
    - Messages are designed purely as data structure
    - To add logic, it is recommended to use wrap classes

- The resulting binary message format is not self-describing
  - Unlike XML, descriptive elements (attribute / element names) are not transmitted
  - The meaning of individual fields can only be determined using the schema
  - The advantage is a significant reduction in message size

© OTE, a.s. 2026 – All rights reserved

What is protobuf

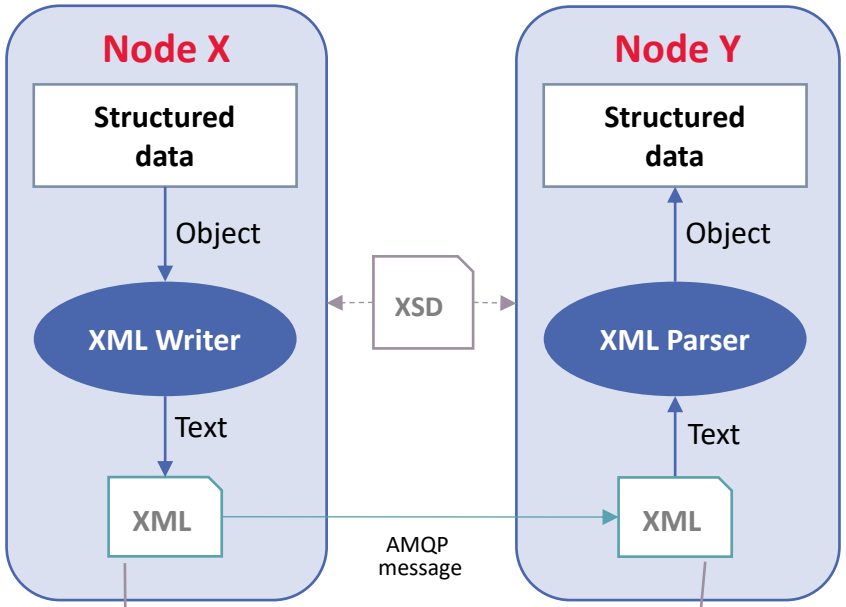
- Designed to be compatible with changes in the data structure
  - Enables easy adaptation to evolving application requirements
  - Requires following recommended best practices when modifying the schema
    - Under these conditions, the format is both backward and forward compatible

- Recommended approach to schema changes
  - Adding a new value to an enum
  - Adding a new (optional) attribute to a message definition
  - Removing an attribute from a message definition (has certain impacts)

© OTE, a.s. 2026 – All rights reserved

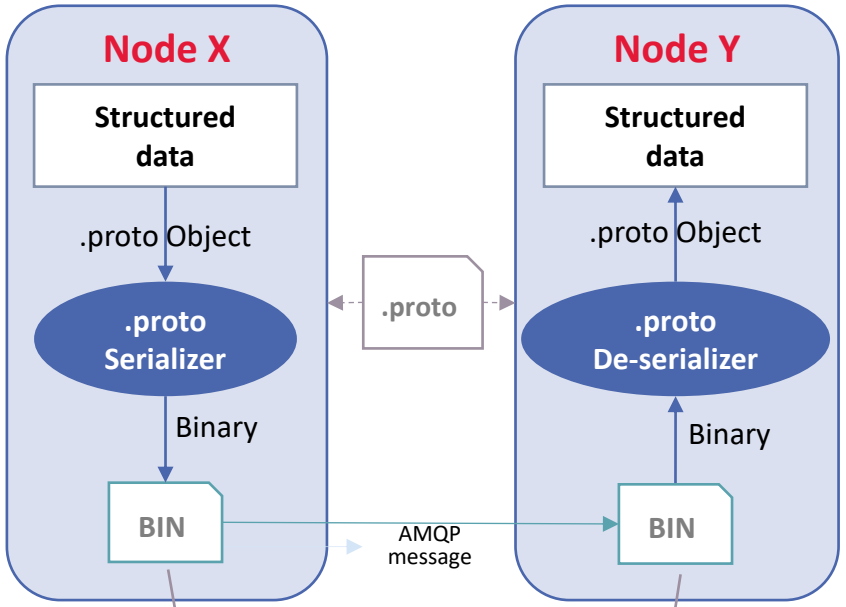
What is protobuf

Existing xml schema




```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PblcOrdrBooksDeltaRprt>
   <StandardHeader marketID="IM"/>
   <OrdrbookList>
      <OrdrBook revisionNo="1" contract="INTRADAY_1H_121221_00_01" dlvryAreaId="DLVRY-AREA-----X"
totalQty="10000" highPx="10000" lowPx="10000">
         <BuyOrdrList>
            <OrdrBookEntry ordrId="123" qty="10000" px="10000" ordrEntryTime="2012-12-20T23:00:00Z"
ordrExeRestriction="NON"/>
         </BuyOrdrList>
      </OrdrBook>
   </OrdrbookList>
</PblcOrdrBooksDeltaRprt>
```

Future protobuf schema




```
00000000  0A 02 08 02  12 4B 0A 18  49 4E 54 52  41 44 41 59  .....K..INTRADAY
00000010  5F 31 48 5F  31 32 31 32  32 31 5F 30  30 5F 30 31  _1H_121221_00_01
00000020  12 10 44 4C  56 52 59 2D  41 52 45 41  2D 2D 2D 2D  ..DLVRY-AREA----
00000030  2D 58 18 01  30 90 4E 48  90 4E 50 90  4E 62 12 08  -X..0.NH.NP.Nb..
00000040  7B 10 90 4E  18 90 4E 22  06 08 F0 B0  CE 86 05 28  {..N..N".......(
00000050  03                                                  .
```

© OTE, a.s. 2026 – All rights reserved

Protobuf 3 Definition

Data Types

- Scalar data types
  - `bool`, `int32`, `int64`, `float`, `double`, `string`, `bytes`, …
- Enumerations
  - A predefined set of values
  - Names must be unique
    - Name prefixing
  - The first value must always represent “unspecified”
    - prefix + `UNSPECIFIED/UNKNOWN`
    - The numeric value must be 0

```
enum MessageType {
  MESSAGE_TYPE_UNSPECIFIED = 0;
  MESSAGE_TYPE_PUBLIC = 1;
  MESSAGE_TYPE_PRIVATE = 2;
}
```

© OTE, a.s. 2026 – All rights reserved

Protobuf 3 Definition

Data Types

- Composite data types
  - Structured definition of a data message
  - Specifies a list of fields along with:
    - cardinality
    - data type
    - attribute name
    - attribute order within the message
      - Attribute order corresponds to the identifier used in binary data

```
message MessageEntry {
  int64 message_id = 1;
  MessageType type = 2;
  optional string contract = 3;
  …
}
```


© OTE, a.s. 2026 – All rights reserved

Protobuf 3 Definition

Field cardinality

- Explicit (`optional`)
  - An explicitly set value is always transmitted
  - An unset value is not transmitted
  - Reintroduced in proto3 to allow detection of whether a default value was explicitly set
- Implicit
  - Object types behave as explicitly `optional`
  - Primitive data types have a default value (`int`: 0, …)
    - Default values are not transmitted
    - During deserialization, it is not possible to determine whether the default value was explicitly set

© OTE, a.s. 2026 – All rights reserved

Protobuf 3 Definition

Field cardinality

- List / repeated field (repeated)
  - Used when a field can have zero or any number of occurrences

```
message RepeatedTest {
  repeated string product_names = 1;
}
```

- Value map (map)
  - Allows defining a typed key/value structure
  - Can contain zero or any number of entries

```
message MapTest {
  map<string, int32> = 1;
}
```


© OTE, a.s. 2026 – All rights reserved

Comparison of Transmitted Data

Current XML interface

- Message content is structured and easily readable
- A simple Public order book update with a single order has a size of 533 bytes
  - With gzip compression, the size is reduced to 320 bytes

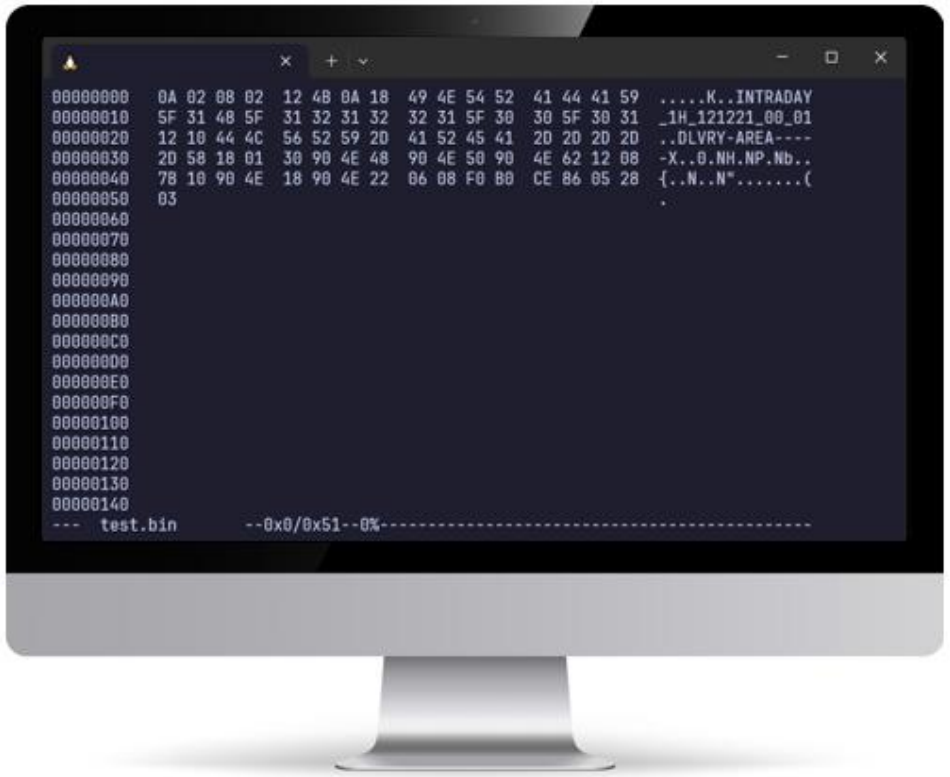
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PblcOrdrBooksDeltaRprt>
    <StandardHeader marketID="IM"/>
    <OrdrbookList>
        <OrdrBook revisionNo="1" contract="INTRADAY_1H_121221_00_01" dlvryAreaId="DLVRY-AREA-----X" totalQty="10000" highPx="10000" lowPx="10000">
            <BuyOrdrList>
                <OrdrBookEntry ordrId="123" qty="10000" px="10000" ordrEntryTime="2012-12-20T23:00:00Z" ordrExeRestriction="NON"/>
            </BuyOrdrList>
        </OrdrBook>
    </OrdrbookList>
</PblcOrdrBooksDeltaRprt>
```


© OTE, a.s. 2026 – All rights reserved

Comparison of Transmitted Data

Future Protobuf interface

- The message is not human-readable at first glance
  - It lacks information about the semantic meaning of individual values
  - Values can only be interpreted with knowledge of the schema
- The same data now has a size of only 81 bytes
  - 15% of the original XML size
  - 25% of the original XML + gzip size
- The transition also includes renaming of values
  - Original names were abbreviations
  - This change does not affect the resulting message size
  - The advantage is improved clarity of the interface



© OTE, a.s. 2026 – All rights reserved

Comparison of Transmitted Data

Specification for the current XML schema

OTECOM (elektřina) - ÚT						
XML Tag name	Tag type	Parent	Data Type	Description	m/o	mult.
MsgRprt	SE		Structure		m	[1..1]
StandardHeader	SE	*MsgRprt*	*Structure*		*m*	*[1..1]*
MsgList	SE	MsgRprt	Structure	List of Orders	o	[0..1]
Msg	SE	OrdrList	Structure		o	[0..n]
msgId	A		Long	The message ID as assigned by the CS OTE system.	m	[1..1]
type	A		String	Defines the message type. Valid Values: "PUBLIC": The message is a public message. "PRIVATE": The message is a private message.	m	[1..1]
contract	A		String	Underlying contract.	o	[0..1]
messageCode	A		Integer	Message code of the message	m	[1..1]
timestmp	A		DateTime	Timestamp of the message as assigned by the CS OTE system.	m	[1..1]
svrty	A		String	Severity of the message: "URG": Urgent message. "ERR": Error. "HIG": High prioritized message. "MED": Medium prioritized message. "LOW": Low priority message.	m	[1..1]
mktSupervisionMsg	A		Boolean	Determines if the message has been send by market supervision	m	[1..1]
txtEn	A		String	Message text. – English version.	m	[1..1]
txtCz	A		String	Message text. – Czech version.	m	[1..1]
sellDlvryAreaId	A		String	In case of an order execution, this field contains the delivery area of the sell side.	o	[0..1]
buyDlvryAreaId	A		String	In case of an order execution, this field contains the delivery area of the buy side.	o	[0..1]

Specification for the protobuf content

OTECOM (elektřina) - ÚT						
Message/Field	Type	Parent	Data Type	Description	m/o	mult.
MessageRprt	MSG		Structure		m	[1..1]
standard_header	FIELD	MessageRprt	*Structure*		*m*	*[1..1]*
~~MsgList~~	~~SE~~	~~MsgRprt~~	~~Structure~~	~~List of Orders~~	~~o~~	~~[0..1]~~
messages	FIELD	MessageRprt	Structure		o	[0..n]
message_id	FIELD		Integer(64)	The message ID as assigned by the CS OTE system.	m	[1..1]
type	FIELD		Enum	Defines the message type. Valid Values: **"MESSAGE_TYPE_PUBLIC"**: The message is a public message. **"MESSAGE_TYPE_PRIVATE"**: The message is a private message.	m	[1..1]
contract	FIELD		String	Related underlying contract (if any).	o	[0..1]
message_code	FIELD		Integer	Message code of the message	m	[1..1]
timestamp	FIELD		Timestamp	Timestamp of the message as assigned by the CS OTE system.	m	[1..1]
severity	FIELD		Enum	Severity of the message: **"MESSAGE_SEVERITY_TYPE_URG"**: Urgent message. **"MESSAGE_SEVERITY_TYPE_ERR"**: Error. **"MESSAGE_SEVERITY_TYPE_HIG"**: High prioritized message. **"MESSAGE_SEVERITY_TYPE_MED"**: Medium prioritized message. **"MESSAGE_SEVERITY_TYPE_LOW"**: Low priority message.	m	[1..1]
market_supervision_message	FIELD		Boolean	Determines if the message has been send by market supervision	m	[1..1]
text_en	FIELD		String	Message text. – English version.	m	[1..1]
text_cz	FIELD		String	Message text. – Czech version.	m	[1..1]
sell_delivery_area_id	FIELD		String	In case of an order execution, this field contains the delivery area of the sell side.	o	[0..1]
buy_delivery_area_id	FIELD		String	In case of an order execution, this field contains the delivery area of the buy side.	o	[0..1]

© OTE, a.s. 2026 – All rights reserved

Protobuf – pecification vs. *.proto* Definition

Protobuf content specification

OTECOM (elektřina) - ÚT						
Message/Field	Type	Parent	Data Type	Description	m/o	mult.
ErrResp	MSG		Structure		m	[1..1]
standard_header	FIELD	ErrResp	*Structure*		*m*	*[1..1]*
errors	FIELD	ErrResp	Structure		m	[1..n]
error_code	FIELD		Integer	Predefined error codes. Some error messages do not have a specific error code. In this case the value is 0.	m	[1..1]
error_en	FIELD		String	The error message for this error – English version.	m	[1..1]
error_cz	FIELD		String	The error message for this error – Czech version.	m	[1..1]
client_order_id	FIELD		String	Client order ID.	o	[0..1]

.proto definition

```
syntax = "proto3";
package cz.ote_cr.otecom.api.ele.v5;

import "google/protobuf/timestamp.proto";
import "google/protobuf/duration.proto";
option java_multiple_files = true;
option java_package = "cz.ote_cr.otecom.api.ele.v5";
import "validate/validate.proto";


message ErrResp {
  StandardHeader standard_header = 1 [(validate.rules).message.required = true];
  repeated ErrorEntry errors = 2;
}

message StandardHeader {
  MarketIdType market_id = 1 [(validate.rules).enum = {not_in: [0], defined_only: true}];
  optional string client_correlation_id = 2;
  optional string client_data_string = 3;
}

message ErrorEntry {
  int32 error_code = 1;
  string error_en = 2;
  string error_cz = 3;
  optional string client_order_id = 4 [(validate.rules).string.max_len = 40];
}
```

OTECOM (elektřina) - protobuf						
Message/Field	Type	Parent	Data Type	Description	m/o	mult.
standard_header	FIELD		Structure		m	[1..1]
market_id	FIELD		Enum	Market Identification Code (MIC) of the market to which the request is sent or from which the request originates. The following values are allowed: "MARKET_ID_TYPE_XBID": XBID Intraday market "MARKET_ID_TYPE_IM": OTE secondary Intraday market (fallback to XBID).	m	[1..1]
client_correlation_id	FIELD		String	The client data field in this section can be used by the client to store	o	[0..1]

© OTE, a.s. 2026 – All rights reserved

Data validation

Data validation

- Protobuf itself does not provide built-in support for validation
  - However, it allows extensibility through plugins

- Validation is implemented using protoc-gen-validate (PGV)
  - Validation rules are defined via attribute annotations
    - Support for individual rules varies depending on the target programming language
  - Compiling the schema with protoc and PGV additionally generates validation code
    - The implementation approach may differ across programming languages
    - For Java, for example, dedicated `*Validator` classes are generated

© OTE, a.s. 2026 – All rights reserved

Data validation

- Mandatory field

```
message AckResp {

  StandardHeader standard_header = 1 [
    (validate.rules).message.required = true
  ];

}
```

- Maximum length

```
message ErrorEntry {

…

  optional string client_order_id = 4 [
    (validate.rules).string.max_len = 40
  ];

}
```

- Enum values

```
message StandardHeader {

  MarketIdType market_id = 1 [(validate.rules).enum = {
    not_in: [0], defined_only: true
  }];

  …

}
```

© OTE, a.s. 2026 – All rights reserved

Changes in the AMQP Interface

Changes in the AMQP Interface

RabbitMQ Message Attributes

- The transition to the new interface requires distinguishing the new type of transmitted messages
- This is achieved by
  - Increasing the version in the *content type* attribute
  - Newly using the *type* attribute, which contains the name of the transmitted message
    - The reason is that the protobuf format itself does not carry information about the message type

Attribute	Current Value	Value for Protobuf
content type	market/{type}; version=4	market/{type}; version=5
content encoding	-/gzip	-/gzip
type	-	*Name of proto message*

© OTE, a.s. 2026 – All rights reserved

Sequence of Broadcast Messages

Message Distribution Sequence

Sequence of Broadcast Messages

- Broadcasted messages include additional information about their order
  - This numeric sequence starts at 1 after the server starts
  - Each routing key maintains its own independent sequence
  - In case of an issue, the sequence allows identification of which data became inconsistent
    - The client tracks the last received sequence number for each unique routing key
    - Receiving a message that does not follow the previous sequence number indicates inconsistency
      - The recovery mechanism is left to the needs of the individual client application
- Limitations of the current approach
  - A missing message can only be identified after successfully receiving a subsequent message
    - The delay depends on the frequency of message broadcast for the given routing key
  - A missing queue binding cannot be distinguished from a general broadcasting error

© OTE, a.s. 2026 – All rights reserved

Message Distribution Sequence

SequenceNumbersRprt

- The solution is to broadcast a special “synchronization” message
  - Automatically broadcasted to all trading participants every 5 seconds
  - Contains a list of all public routing keys broadcasted up to that moment
  - For each routing key, it records the last sent sequence number

- Provides a faster and more reliable way to detect broadcasting issues
  - Reduces uncertainty about missing messages to a short time window
  - Ideally, both approaches should be combined
    - Gap in sequence monotonicity in each received message identifies sporadic outage
    - Validation using SequenceNumbersRprt identifies longer-lasting outage

© OTE, a.s. 2026 – All rights reserved

Digital Signature

Digital signature

- Electronic signature security remains preserved for the same messages as in the original interface
  - `ModifyOrderReq`
  - `AddOrderReq`
  - `ModifyAllOrdersReq`

- However, the approach of embedding the digital signature directly into the XML message content is no longer possible in the protobuf interface

© OTE, a.s. 2026 – All rights reserved

Digital Signature

Current XML interface

- The digital signature is embedded as a new element inside the transmitted message
  - This simplifies distribution, as the message does not need to be „additionally wrapped“ for further distribution

```
<?xml version="1.0" encoding="UTF-8"?>
<OrdrEntry>
    <StandardHeader marketID="IM"/>
    <OrdrList>
        <Ordr contract="INTRADAY_1H_150701_15_16" px="10000" qty="10000" side="BUY" ... />
    </OrdrList>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
...
</ds:Signature></OrdrEntry>
```


© OTE, a.s. 2026 – All rights reserved

Digital Signature

Future proto interface

- The digital signature is not part of the protobuf specification
- Protobuf itself does not have a canonical message form
  - The same message can be serialized in multiple ways
  - With XML, the message could be reformatted and the signature would remain valid
- Messages must therefore be signed using a different standardized approach
  - The new format is based on the CMS standard defined in RFC 5652
  - The result of signing is a SignedData ASN.1 structure containing
    - Original message, signatue, information about the used certificate

© OTE, a.s. 2026 – All rights reserved

Digital Signature

- AddOrderReq is converted into the Protobuf format
- It is signed and converted into the CMS format
- These data are placed into a SignedMessage together with:
  - Information about the type of the signed message
  - The original content encoding
- The resulting message is converted back into the Protobuf format
- RabbitMQ message attributes are set:
  - Message type is SignedMessage
  - Content encoding of the signed message
  - Content type with the new interface version

© OTE, a.s. 2026 – All rights reserved

The next steps in Binary API implementation:

- **Webinar on Binary API** (02.06.2026 at 14:00)
  - Send any questions to market@ote-cr.cz
  - The presentation will be published on the OTE website
  - We will continuously publish a document with any questions and answers

- **Start of testing with Market Participants (September/October 2026)**
  - You will be informed in advance about the details of the start of testing

- **Launch of Binary API under SIDC and OTE (Q1 2027)**

© OTE, a.s. 2026 – All rights reserved

Thank you for your attention

Ing. Jakub Šrom, Ph.D.,
jsrom@ote-cr.cz

Ing. Lukáš Vitásek