

Zavedení Binary API na trzích OTE

Jakub Šrom, Lukáš Vitásek
2. 6. 2026



Představení změny rozhraní

Harmonogram a trendy

Business oblast

Co je to protobuf

Validace dat

Změny v AMQP rozhraní

Sekvence při distribuci zpráv

Digitální podpis

Technická oblast

Představení změny rozhraní

Změna rozhraní automatické komunikace na Binary API

- **V rámci projektu SIDC (Single Intraday Coupling) pro zajištění odpovídajícího výkonu a stability centrálního systému XBID (změna rozhraní OTE-COM – XBID)**
- **XML formát zpráv bude nahrazen binárním obsahem, datovým formátem Protocol Buffers.**
- Koncept komunikace (stávající komunikační scénáře a komunikační protokol AMQP) se z pohledu přechodu na nové komunikační scénáře nemění.
- **Pro zajištění benefitů výkonnější a rychlejší komunikace pro účastníky trhu OTE nutno adaptovat rozhraní AK na zamýšlený binární formát Protocol Buffers i na straně CS OTE (AMQP komunikace)**

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

Úprava automatické komunikace AMQP na:

- Vnitrodenním kontinuálním trhu s elektřinou
- Vnitrodenním trhu s plynem

→ **XML formát** zpráv bude nahrazen binárním datovým **formátem Protocol Buffers**.

Úprava nemá dopad na:

- uživatele aplikace OTE-COM pro Vnitrodenní trh s elektřinou kontinuální
- uživatele aplikace OTE-COM pro Vnitrodenní trh s plynem
- mobilní aplikace
- CS Portál - krátkodobé trhy (DT a IDA) a evidenci realizačních diagramů

Ke změně komunikačního formátu dojde v rámci automatické komunikace VDT i VDP vůči účastníkům trhu.

Poskytnutá dokumentace zahrnuje :

Aktualizovaná verze manuálu formátů zpráv elektřina

Aktualizovaná verze manuálu formátů zpráv plyn

XLSX mapovací tabulky, které poskytují mapování stávajících názvů XML atributů na nové názvy BIN API ekvivalentů těchto atributů

- XLSX mapovací tabulky elektřina
- XLSX mapovací tabulky plyn

Protobuf definice (.proto soubory):

- Za elektřinu (VDT)
- Za plyn (VDP)

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 Formátv zprav Binary API OTE-ODM ELE verzeC D1.4.X Formátv zprav Binary API OTE-ODM 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-ODM ele protobuf vs XML verzeB (zůstává kompatibilní s verzí C Formátů zpráv elektřina)
Protobuf definice (.proto soubor) elektřina	OTE-ODM ele protobuf
Formáty zpráv plyn	D1.5.1 Formátv zprav Binary API OTE-ODM GAS verzeB D1.5.1 Formátv zprav Binary API OTE-ODM 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 OTE-ODM gas protobuf vs XML verzeB
Protobuf definice (.proto soubor) plyn	OTE-ODM gas protobuf

→ Ke stažení na dedikované stránce k Binary API na webu [OTE zde](#)

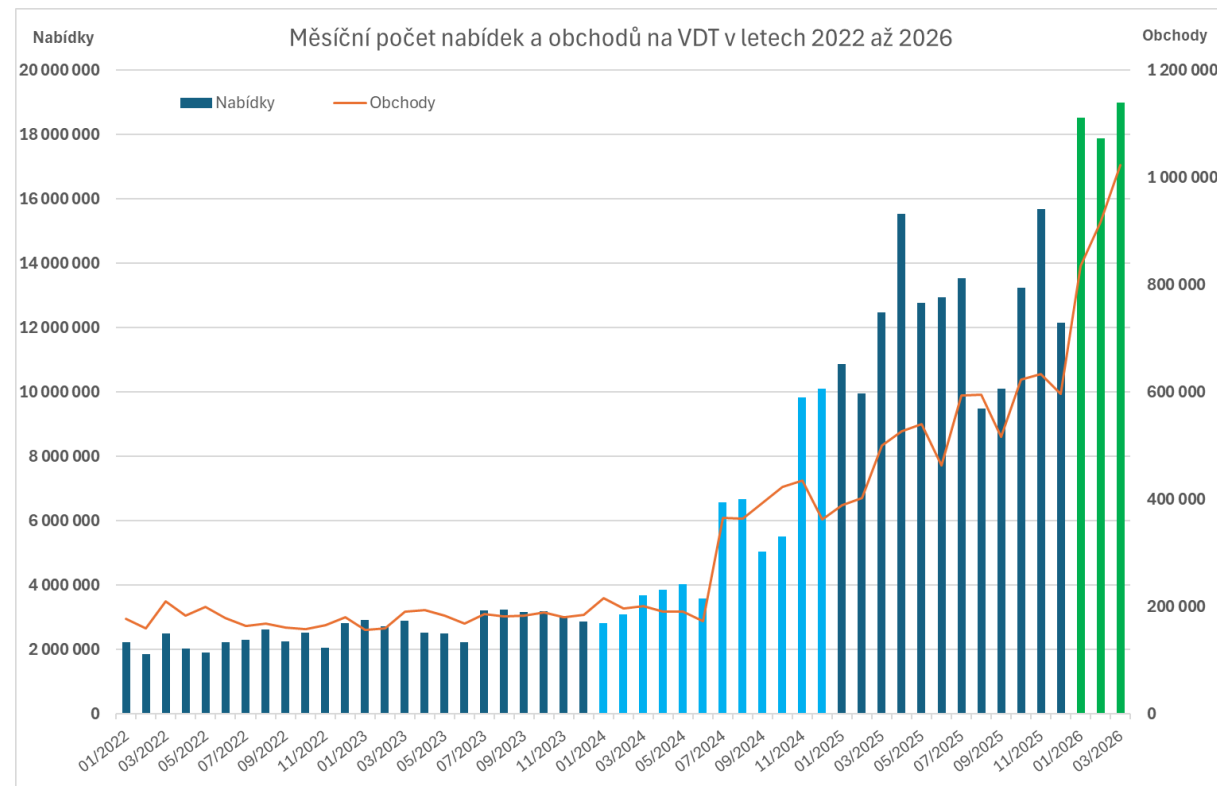
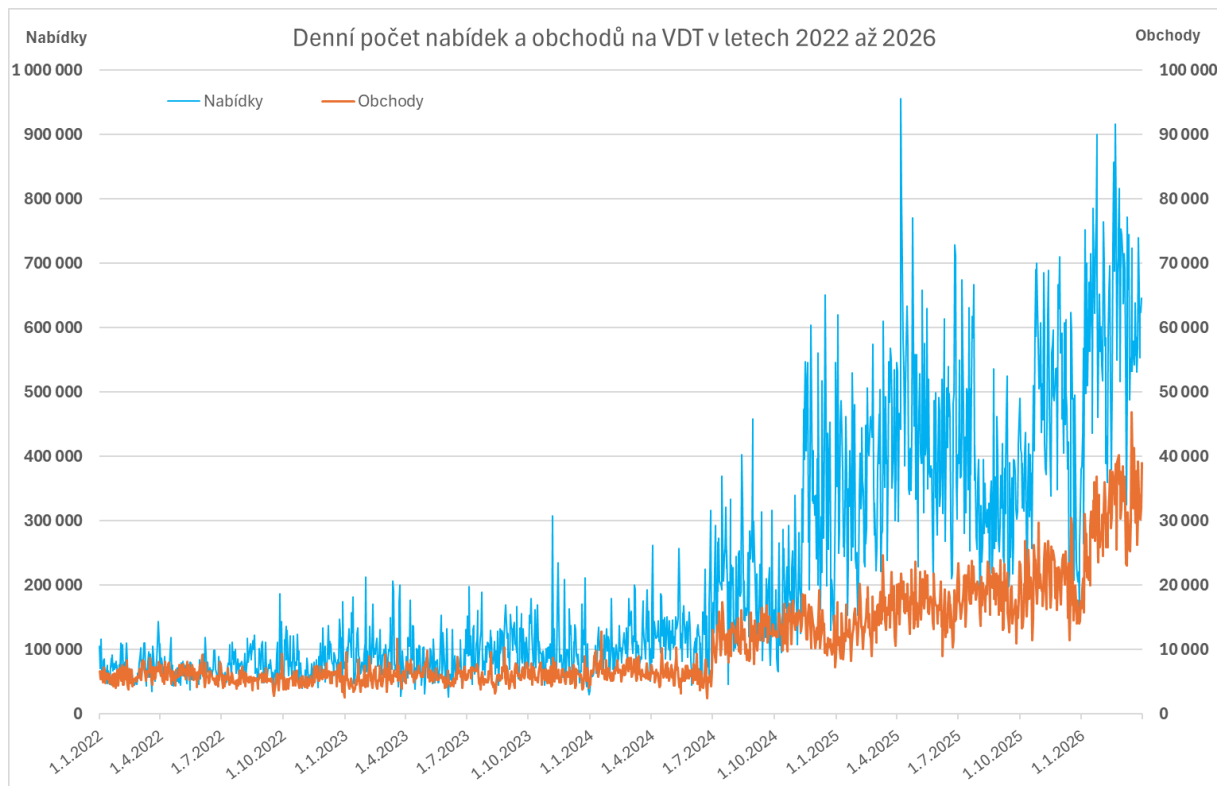
Harmonogram a trendy

Harmonogram v implementaci Binary API:

- Prvotní informace na účastníky trhu (17.12.2025)
- Zveřejnění Protobuf definicí (.proto soubory):
 - Za elektřinu (VDT) (21.4.2026)
 - Za plyn (VDP) (25.05.2026)
- Webinář k Binary API (02/06/2026 ve 14:00)
 - Případné dotazy zasílejte na **market@ote-cr.cz**
 - Prezentace bude zveřejněna na webu OTE
 - Průběžně budeme zveřejňovat dokument s případnými dotazy a odpověďmi
- **Zahájení testování s ÚT (září/říjen 2026)**
 - O detailech k zahájení testování budete v předstihu informováni
- **Spuštění Binary API v rámci SIDC a OTE (Q1 2027)**

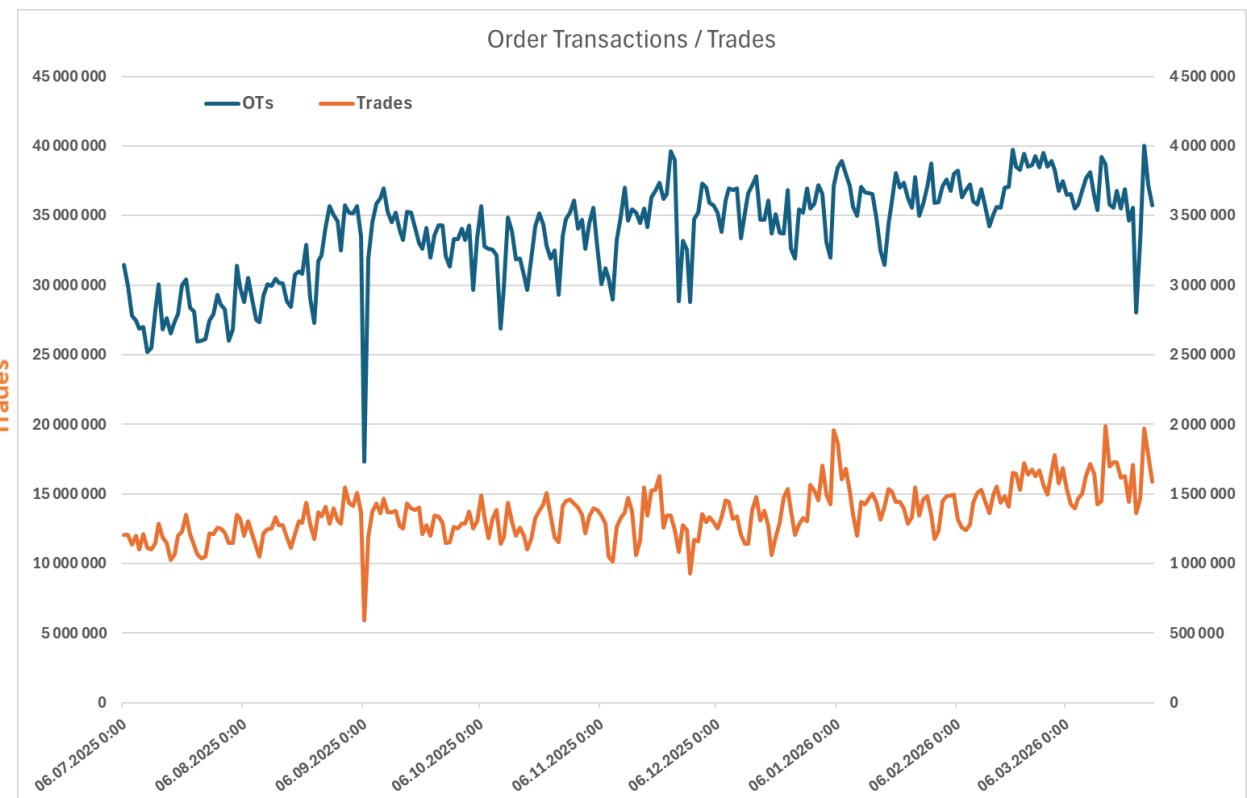
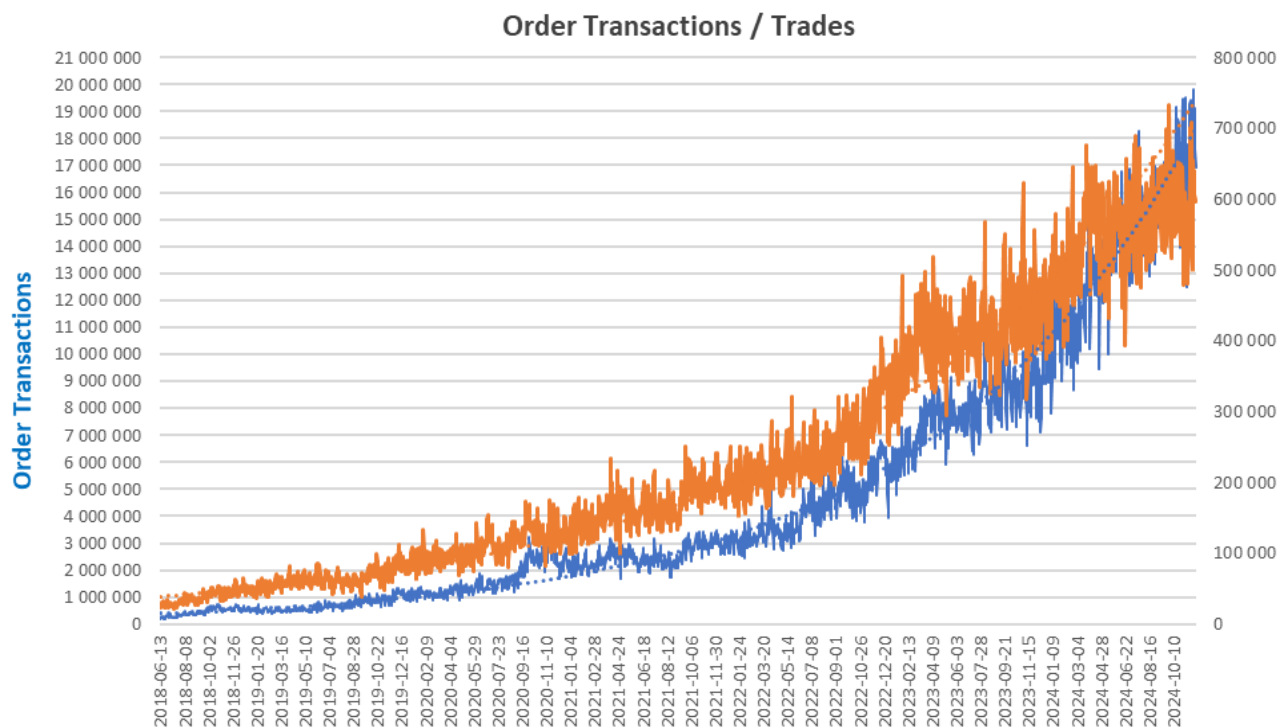
OTE COM

- Trend algoritmického obchodování v celé Evropě – zapojování automatů
- Strmý nárůst nabídek vs uzavírané obchody
- Více obchodování blízko reálnému času



Vývoj v SIDC

- Denní počty nabídek a obchodů v rámci SIDC od prvního obchodního dne (13. června 2018)



SIDC – Single Intraday Coupling – Jednotný vnitrodenní trh s elektřinou

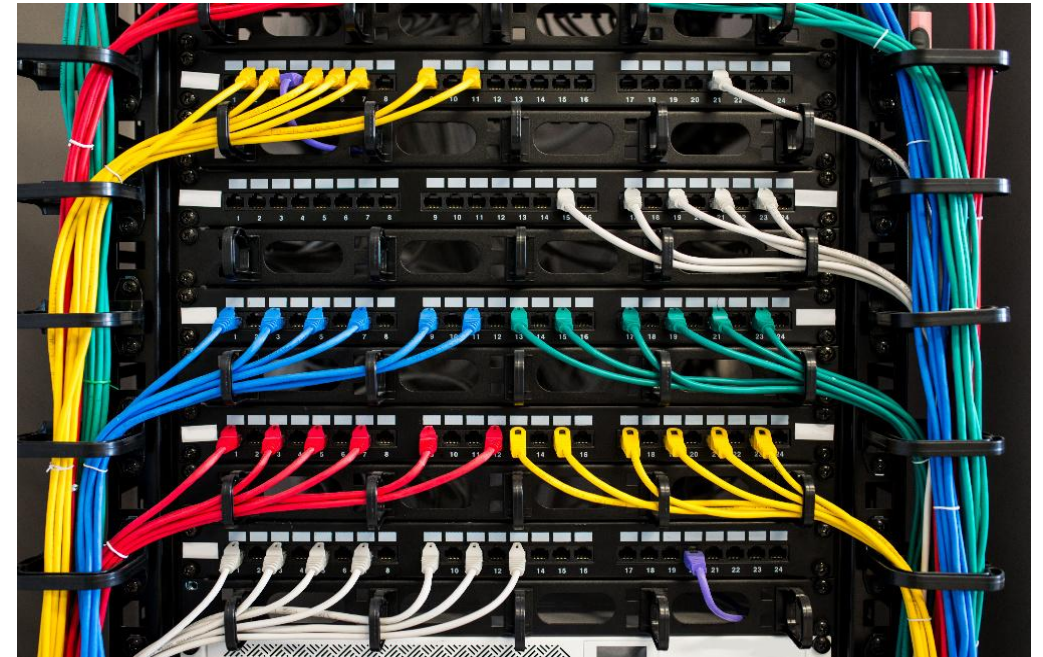
Co je to protobuf

Proč tato změna?

- transakční zátěž v rámci SIDC v posledních letech významně roste
 - jednak je zřejmý obecný trend nárůstu obchodování
 - a navíc zavedení 15min produktů přineslo významný nárůst počtu kontraktů
 - s tím úměrně narostl počet distribuovaných změn (nejen) vývěsky
- současný způsob přenosu zpráv naráží na své limity
 - (de)serializace a validace XML je časově náročná
 - XML samotné je „objemově náročný“ datový formát
 - zprávy jsou větší, než je nezbytně nutné
- XBID se tento problém snaží řešit přechodem na protobuf formát

Co je to protobuf

- datový formát vyvinutý společností Google určený pro serializaci strukturovaných dat do binární podoby
- při návrhu kladen důraz na jednoduchost a výkonost
- v porovnání s jinými formáty, jako například JSON a XML, nabízí 2-4x menší velikost zpráv
- web projektu: <https://protobuf.dev/>



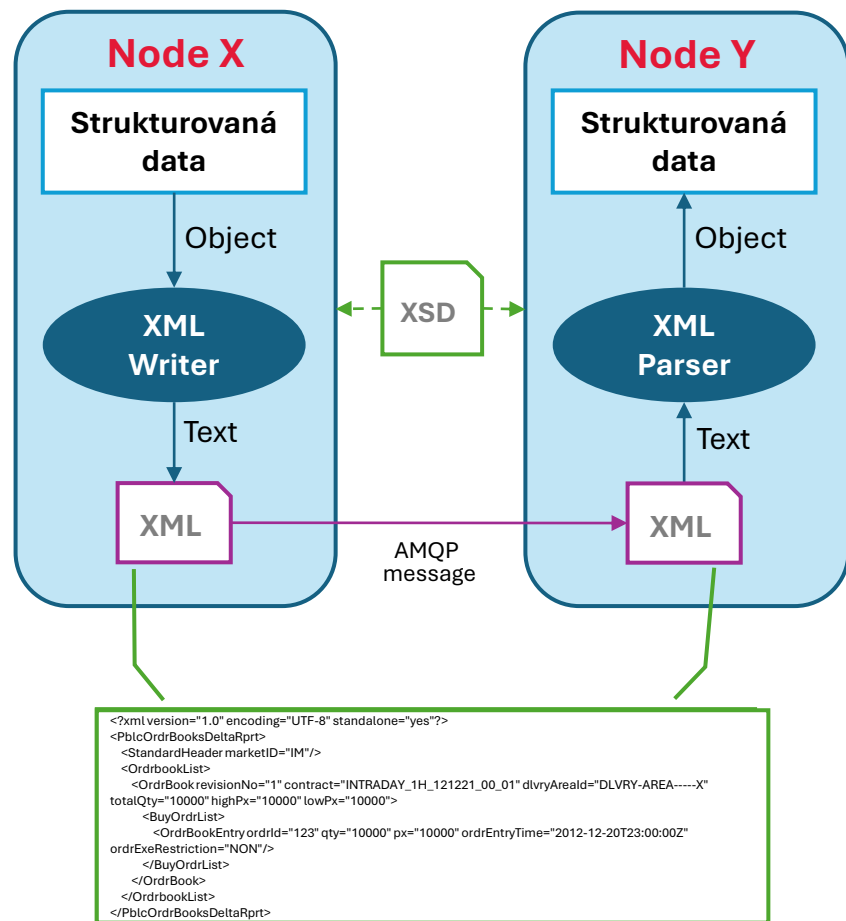
Co je to protobuf

- struktura zpráv je definována v textovém .proto schématu
 - obdoba XSD, definuje samotné zprávy a jejich strukturu
 - ve výchozím stavu naopak neumí validační pravidla
 - schéma je kompilováno pomocí protoc nástroje do zdrojového kódu
 - zprávy jsou navrženy pouze jako datová struktura
 - pro přidání logiky je doporučováno třídy například zaobalit
- výsledný binární formát zpráv není samostatně popisný
 - na rozdíl od XML nejsou popisné položky (názvy atributů / elementů) přenášeny
 - význam jednotlivých položek lze zjistit až s pomocí schématu
 - výhodou je značná úspora ve velikosti zpráv

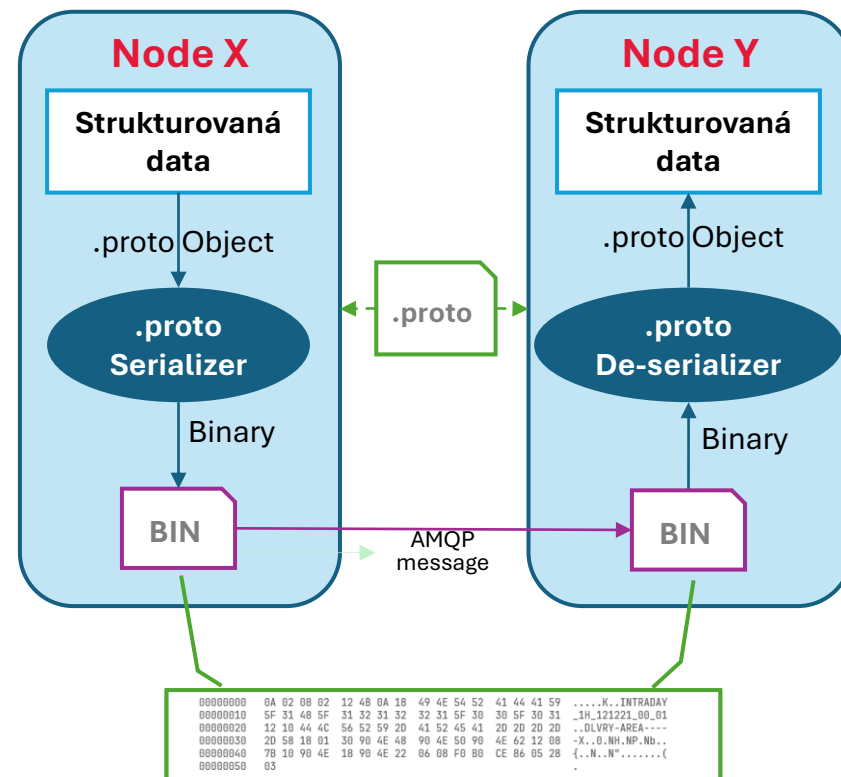
Co je to protobuf

- navržen tak, aby byl kompatibilní se změnami v datové struktuře
 - umožňuje snadné přizpůsobení měnícím se požadavkům aplikace
 - vyžaduje dodržet vhodné postupy pro provádění změn ve schématu
 - za těchto podmínek je formát zpětně i dopředně kompatibilní
- doporučený přístup ke změnám schématu
 - přidání nové položky v enum
 - přidání nového (nepovinného) atributu do definice zprávy
 - odebrání atributu z definice zprávy (má určité dopady)

Stávající XML schéma



Budoucí protobuf schéma



Protobuf 3 definice

Datové typy

- skalární datové typy
 - bool, int32, int64, float, double, string, bytes, ...
- enumerace
 - předem definovaná množina hodnot
 - názvy musí být unikátní
 - prefixing názvu
 - první hodnota je vždy „neurčená“
 - prefix + UNSPECIFIED/UNKNOWN
 - číselná hodnota musí být 0

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

Protobuf 3 definice

Datové typy

- složené datové typy
 - strukturovaný popis datové zprávy
 - uvádí seznam položek spolu s
 - kardinalitou
 - datovým typem
 - názvem atributu
 - pořadím atributu ve zprávě
 - odpovídá identifikátoru v binárních datech

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

Protobuf 3 definice

Kardinalita položek

- explicitní (`optional`)
 - explicitně nastavená hodnota se vždy přenáší
 - nevyplněná hodnota se nepřenáší
 - později vráceno do proto3 z důvodu identifikace nastavení výchozí hodnoty
- implicitní
 - objektové typy se chovají jako explicitní `optional`
 - primitivní datové typy mají výchozí hodnotu (`int: 0, ...`)
 - výchozí hodnota se nepřenáší
 - při deserializaci není zjistitelné, co výchozí hodnota znamenala

Protobuf 3 definice

Kardinalita položek

- seznam / opakovaná položka `repeated`
 - použití v případech, kdy hodnota může žádný nebo libovolný počet výskytů
- mapa hodnot `map`
 - umožňuje definovat (typovou) strukturu klíč/hodnota
 - může mít opět žádný nebo libovolný počet záznamů

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

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

Porovnání přenášených dat

Současné XML rozhraní

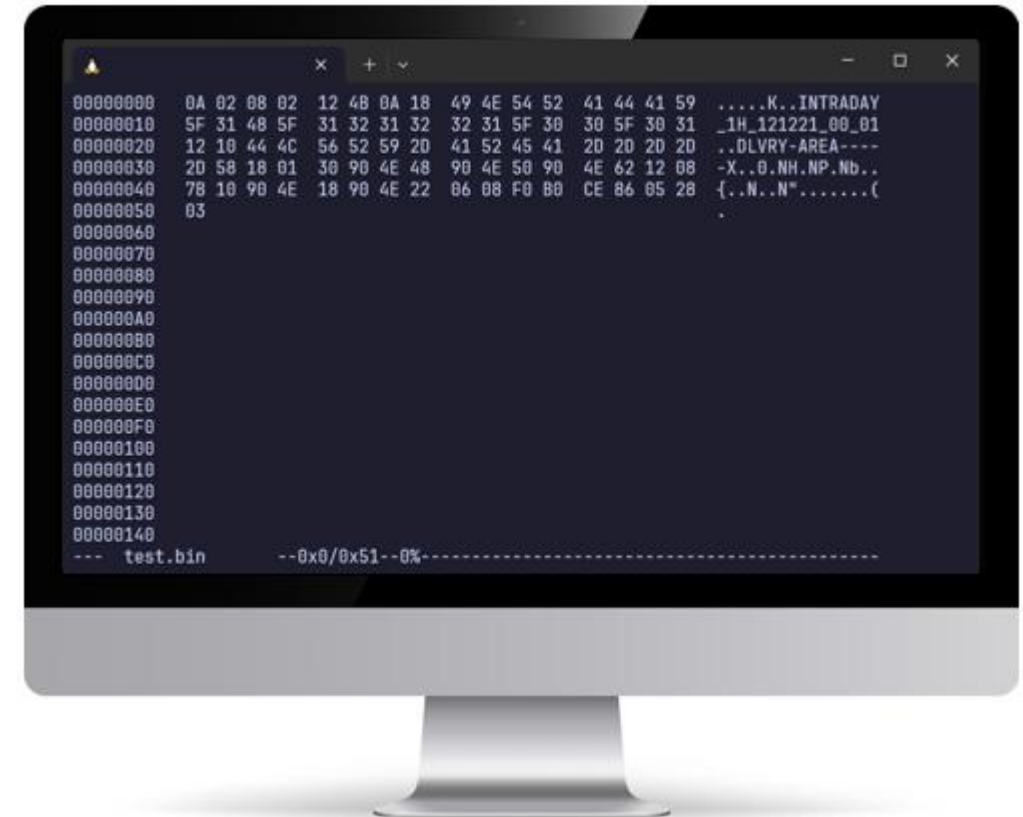
- obsah zpráv je strukturovaný a snadno čitelný
- jednoduchá změna vývěsky s jednou nabídkou má 533 bajtů
 - s použitím gzip komprimace 320 bajtů

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

Porovnání přenášených dat

Budoucí proto rozhraní

- zpráva je na první pohled nečitelná
 - postrádá informace o významu hodnot
 - lze přiřadit až se znalostí schématu
- stejná data mají nyní pouhých 81 bajtů
 - 15% původní velikosti XML
 - 25% původní velikosti XML+gzip
- součástí přechodu je i přejmenování hodnot
 - původní názvy byly zkráceny
 - změna nemá vliv na výslednou velikost
 - výhodou je zpřehlednění rozhraní



Porovnání přenášených dat

Specifikace pro stávající XML schéma

OTECOM (elektrina) - 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]
msgld	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]
timestamp	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]
sellDlrvyAreald	A		String	In case of an order execution, this field contains the delivery area of the sell side.	o	[0..1]
buyDlrvyAreald	A		String	In case of an order execution, this field contains the delivery area of the buy side.	o	[0..1]

Specifikace pro protobuf obsah

OTECOM (elektrina) - 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]

Protobuf – specifikace vs .proto definice

Specifikace protobuf obsahu

OTECOM (elektrina) - ÚT						
Message/Field	Type	Parent	Data Type	Description	m/o	mult.
ErrResp	MSG		Structure		m	[1..1]
<u>standard_header</u>	FIELD	ErrResp	Structure		m	[1..1]
<u>errors</u>	FIELD	ErrResp	Structure		m	[1..n]
<u>error_code</u>	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]
<u>error_en</u>	FIELD		String	The error message for this error – English version.	m	[1..1]
<u>error_cz</u>	FIELD		String	The error message for this error – Czech version.	m	[1..1]
<u>client_order_id</u>	FIELD		String	Client order ID.	o	[0..1]

OTECOM (elektrina) - protobuf						
Message/Field	Type	Parent	Data Type	Description	m/o	mult.
<u>standard_header</u>	FIELD		Structure		m	[1..1]
<u>market_id</u>	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]
<u>client_correlation_id</u>	FIELD		String	The client data field in this section can be used by the client to store information about the client.	o	[0..1]

.proto definice

```

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];
}
    
```

Validace dat

Validace dat

- samotný protobuf nemá zabudovanou podporu pro validaci
 - umožňuje ale rozšířitelnost pomocí pluginů
- pro validaci se využívá protoc-gen-validate (PGV)
 - skrze anotace atributů umožňuje definovat validační pravidla
 - podpora pravidel se liší v závislosti na cílovém programovacím jazyku
 - kompilace schématu přes protoc s PVG vygeneruje navíc validační kód
 - způsob se může lišit pro jednotlivé programovací jazyky
 - pro Java jsou to například *Validator třídy

Validace dat

- povinnost položky

```
message AckResp {  
    StandardHeader standard_header = 1 [  
        (validate.rules).message.required = true  
    ];  
}
```

- maximální délka

```
message ErrorEntry {  
    ...  
    optional string client_order_id = 4 [  
        (validate.rules).string.max_len = 40  
    ];  
}
```

- hodnoty z enum

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

Změny v AMQP rozhraní

Atributy RabbitMQ zpráv

- přechod na nové rozhraní s sebou nese nutnost rozlišení nového typu přenášených zpráv
- to je realizováno
 - navýšením verze v atributu content type
 - nově také využitím atributu type, který obsahuje název přenášené zprávy
 - důvodem je fakt, že formát protobuf sám o sobě nepřenáší informaci o typu zprávy

Atribut	Původní hodnota	Nová hodnota
content type	market/{type}; version=4	market/{type}; version=5
content encoding	-/gzip	-/gzip
type	-	název proto zprávy

Sekvence při distribuci zpráv

Sekvence při distribuci zpráv

- hromadně distribuované zprávy obsahují doplňující informaci o jejich pořadí
 - tato číselná sekvence začíná od 1 po startu serveru
 - každý routovací klíč počítá svou vlastní sekvenci
 - v případě problému umožňuje identifikovat, která data se stala nekonzistentní
 - klient si hlídá poslední sekvenční číslo pro každý unikátní routovací klíč
 - příjem zprávy, která nenavazuje na předchozí hodnotu, indikuje nekonzistenci
 - způsob obnovy je ponechán na potřebách vlastní klientské aplikace
- současný přístup má svá omezení
 - chybějící zpráva je identifikovatelná až s úspěšným příjmem některé následující
 - prodleva záleží na četnosti rozesílání zpráv s daným routovacím klíčem
 - neexistující binding na frontě není rozlišitelný od obecné chyby při distribuci

SequenceNumbersRprt

- řešením je distribuce speciální „synchronizační“ zprávy
 - automaticky distribuována na všechny ÚT v intervalu 5s
 - obsahuje seznam všech do dané chvíle rozeslaných veřejných routovacích klíčů
 - ke každému klíči eviduje poslední odeslané sekvenční číslo

- rychlejší a spolehlivější způsob zjištění problému v distribuci
 - eliminuje nejistotu v chybějících zprávách na krátké časové okno
 - ideální je samozřejmě kombinace obou přístupů
 - monotónnost sekvence u každé přijaté zprávy identifikuje nahodilý výpadek
 - kontrola pomocí SequenceNumbersRprt identifikuje déle trvající výpadek

Digitální podpis

Digitální podpis

- zabezpečení elektronickým podpisem zůstává zachováno u stejných zpráv, jako v původním rozhraní
 - `ModifyOrderReq`
 - `AddOrderReq`
 - `ModifyAllOrdersReq`
- způsob, kterým se digitální podpis vkládal do obsahu samotné XML zprávy, už ale není v protobuf rozhraní možný



Digitální podpis

Současné XML rozhraní

- digitální podpis je vložen jako nový element uvnitř přenášené zprávy
 - usnadňuje distribuci, není ji třeba „zaobalovat“ pro další distribuci

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

Digitální podpis

Budoucí proto rozhraní

- digitální podpis není součástí protobuf specifikace
- samotný protobuf nemá kanonickou formu zpráv
 - stejná zpráva může být serializovaná více způsoby
 - u XML bylo možné přeformátovat zprávu a podpis byl stále platný
- zprávy je potřeba podepisovat jiným standardizovaným způsobem
 - nový formát vychází ze standardu CMS, definovaném v RFC 5652
 - výsledek podpisu je SignedData ASN.1 struktura
 - původní zpráva, podpis, informace o použitém certifikátu

Digitální podpis

- AddOrderReq se převede do proto formátu
- podepíše se a převede do CMS formátu
- tato data se vloží do SignedMessage spolu
 - informací o typu podepsané zprávy
 - původním content encoding
- výsledná zpráva se převede do proto formát
- nastaví se atributy RabbitMQ zprávy
 - typ přenášené zprávy je SignedMessage
 - content encoding podepsané zprávy
 - content type s novou verzí rozhraní



Další kroky v implementaci Binary API:

- Webinář k Binary API
 - Případné dotazy zasílejte na **market@ote-cr.cz**
 - Prezentace bude zveřejněna na webu OTE
 - Průběžně budeme zveřejňovat dokument s případnými dotazy a odpověďmi
- **Zahájení testování s ÚT (září/říjen 2026)**
 - O detailech k zahájení testování budete v předstihu informováni
- **Spuštění Binary API v rámci SIDC a OTE (Q1 2027)**

DĚKUJI ZA POZORNOST

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

Ing. Lukáš Vitásek