

Standard OPC

Procesna avtomatika

Uroš Lotrič, Nejc Ilc

Standard OPC

- prej: **O**bject linking and embedding for **P**rocess **C**ontrol (OPC Classic)
- danes: **O**pen **P**latform **C**ommunication (OPC UA)

Standard interoperabilnosti (medobratovalnosti) za varno in zanesljivo izmenjavo podatkov v industrijski avtomatizaciji. Je neodvisen od platforme in zagotavlja pretok podatkov med napravami različnih proizvajalcev.

Standard OPC je največja izboljšava na področju avtomatizacije po standardu IEC 61131 (standardizirani programski jeziki)

Razvoj in vzdrževanje standarda je odgovornost [OPC Foundation](https://www.opcfoundation.org/)

Uvod (nad.)

Model komunikacije:

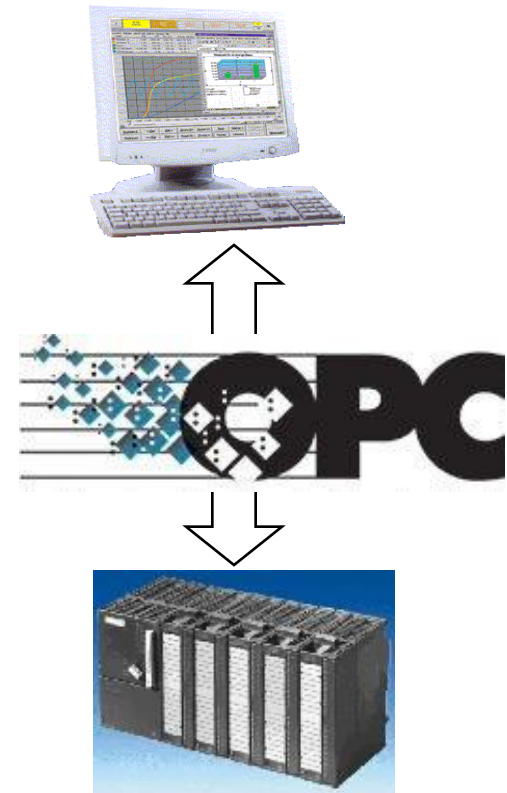
- odjemalec – strežnik
- objava – naročanje (PubSub = publish-subscribe), v verziji 1.04

Strežnik OPC

- Posebnosti naprav so skrite za strežnikom OPC, ki
 - se izvaja na isti ali drugi napravi;
 - jih (strežnike) pripravijo proizvajalci procesnih naprav ali neodvisne programske hiše.
- Strežnik OPC lahko hkrati upravlja več naprav istega tipa
- Hkrati se lahko izvaja več strežnikov OPC

Odjemalec OPC

- Branje in pisanje procesnih podatkov
- Branje in potrjevanje alarmov
- Spremljanje dogodkov
- Pridobivanje podatkov iz arhivske podatkovne baze na podlagi različnih kriterijev



Uvod (nad.)

Prednosti za uporabnika

Transparenten dostop do relevantnih informacij

- Spremenljivke, podatkovni tipi, funkcijski bloki, strukture, ...
- Neodvisno od sistema in omrežja

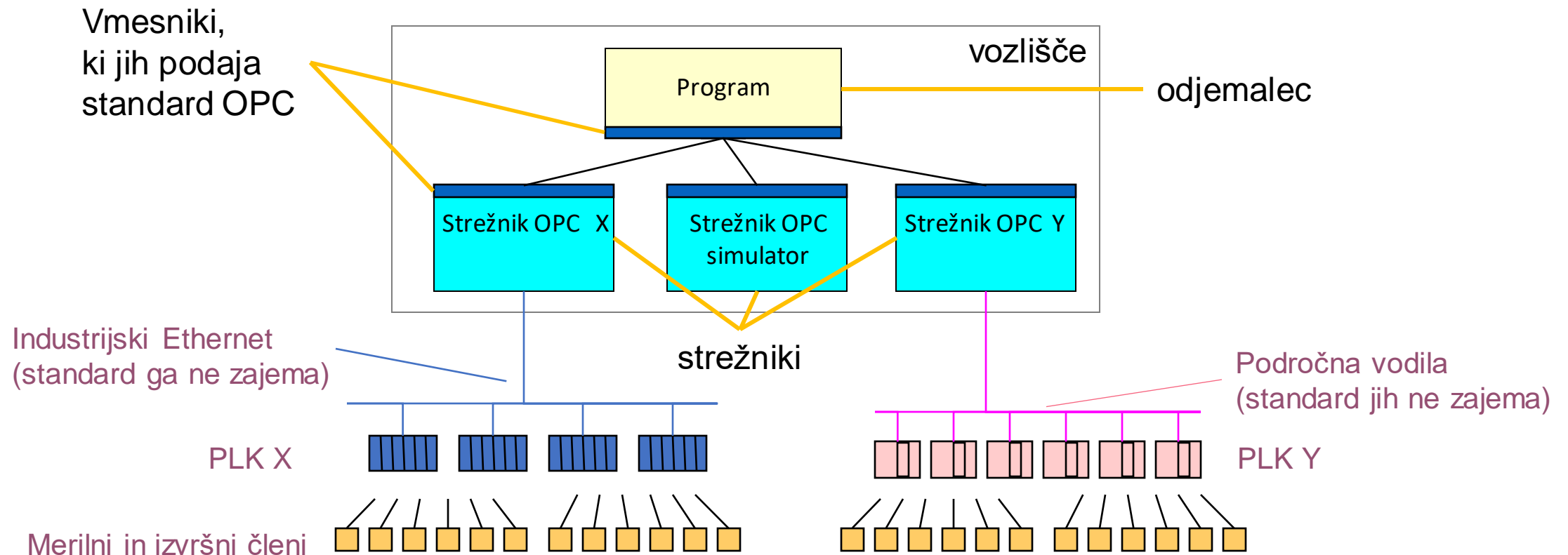
Zmanjša čas integracije

- Krmilnik, HMI, SCADA, ERP, ...

Uvod (nad.)

Standard podaja programski vmesnik (objekte in metode)
za strežnike in odjemalce

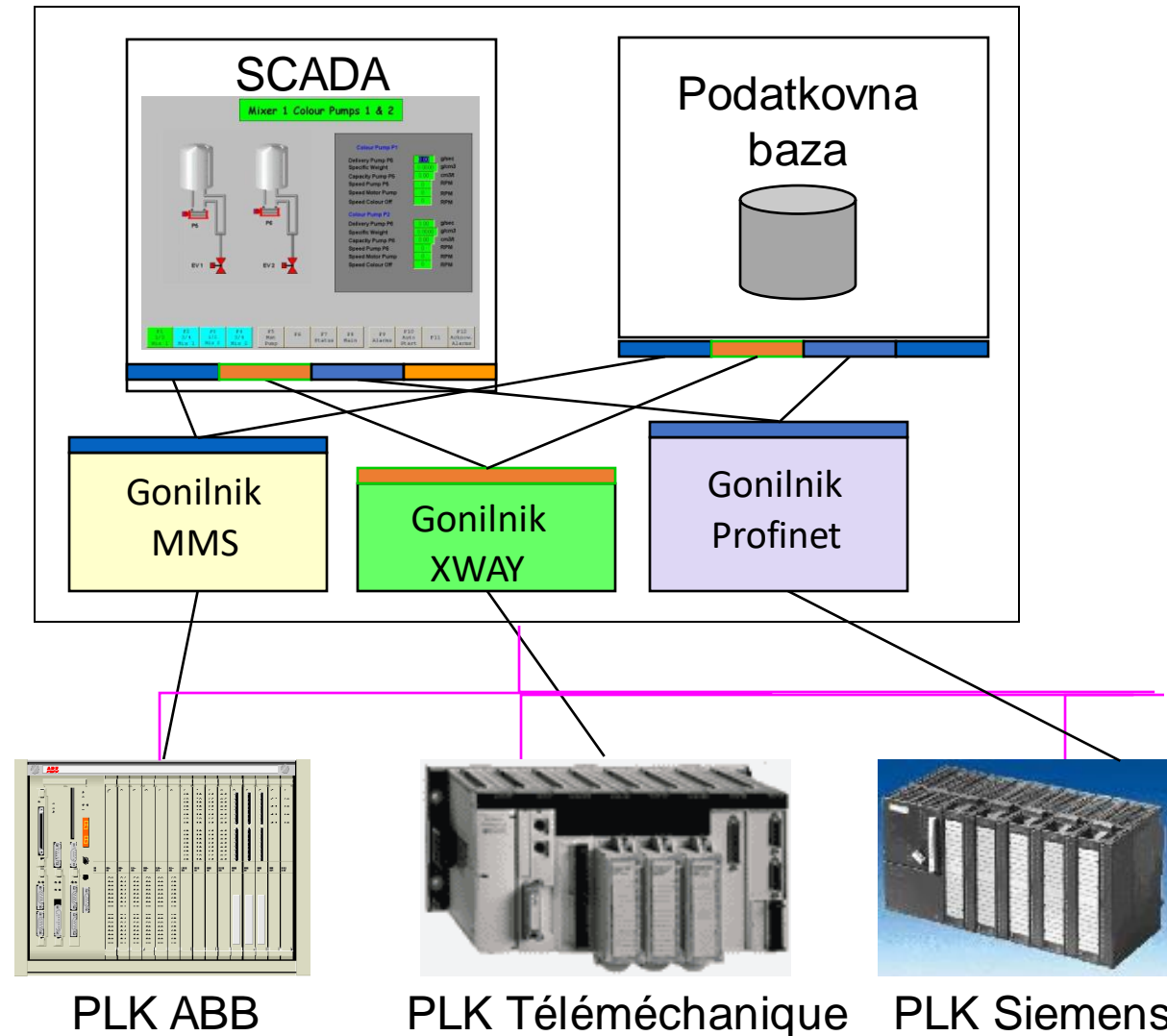
Standard ne pokriva povezave strežnika in naprav za avtomatizacijo



Uvod (nad.)

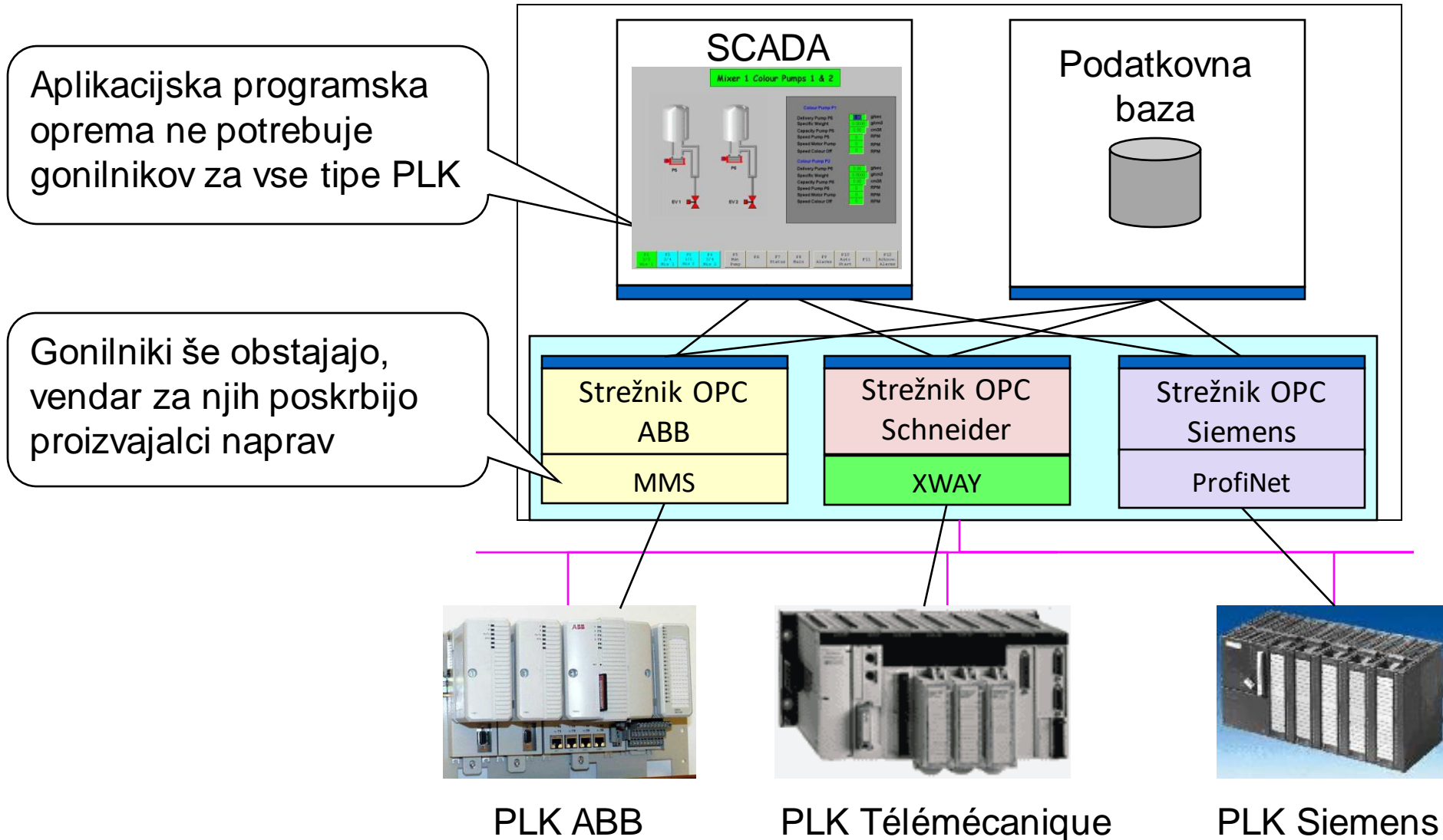
Brez standarda OPC

- Vsak višji sistem vodenja ima poseben vmesnik za dostopanje do gonilnikov naprav



Uvod (nad.)

Standard OPC



Razvoj

Kratka zgodovina razvoja standarda

- **1995:** proizvajalci opreme za avtomatiko (Fischer-Rosemount, Intellution, Opto 22, Rockwell Software) oblikujejo skupino za razvoj standarda
- **1996:** izdana prva verzija OPC, ki se sedaj imenuje **OPC Classic**; temelji na Microsoftovi tehnologiji: Windows + (D)COM
- **2004:** Microsoft izda Windows XP Service Pack 2, ki zelo omeji protokol DCOM in posledično pristriže peruti standardu OPC
- **2006:** izdana prva verzija nove arhitekture **OPC UA** (unified architecture), ki poudarja neodvisnost od tehnologije (OS, transportni protokol, računalniška arhitektura) in varnost (certifikati)
- **Nedavno:** november 2022: izdana specifikacija OPC UA Field eXchange (UAFX), ki razširja uporabo standarda na domeno polja PLKjev ([predstavitveni video](#))

Tehnološki gradniki standarda OPC Classic (še vedno precej prisotni)

- DA, AE, HDA, Batch, DX, XML-DA

OPC Classic – gradniki (nad.)

OPC Classic je opustel/zastarel/zgodovina

OPC-AE (ang. Alarms and Events) za alarme in dogodke

- Alarmi so nenormalna stanja sistema, ki zahtevajo pozornost operaterja
 - Primer: prenizek pritisk olja
- Dogodki so spremembe v procesu, ki se vpisujejo v dnevnik
 - Primer: začetek proizvodnje
- OPC-AE določa
 - Zbiranje alarmov in dogodkov (čas dogodka)
 - pod kakšnimi pogoji (filtriranje, prioritete) so odposlani
 - Potrjevanje alarmov
 - Povezovanje alarmov s tekstovnimi opisi
- Odjemalci OPC-AE se največ uporabljajo za dnevnik alarmov in dogodkov

The screenshot displays the TeMIPClient application window. The main area shows a table of alarms and events with columns for S, P, A, Event Type, Perceived Severity, Probable Cause, and Managed Object. The table contains several rows of data, including 'ProcessingError', 'Environment', and 'AdapterError' with various severity levels like Warning, Critical, and Major. Below the table, there are statistics for filtered alarms: 'Filtered Alarms (Total): 9' and 'Filtered Alarms (New): 22'. At the bottom, there are two smaller tables: one for monitored objects and another for event history.

S	P	A	Event Type	Perceived Severity	Probable Cause	Managed Object
Warning			ProcessingError	Warning	StorageCap...	OPERATIO...
Critical			ProcessingError	Critical	StorageCap...	OPERATIO...
Critical			ProcessingError	Critical	StorageCap...	OPERATIO...
Major			Environment...	Major	CallEstablis...	OPERATIO...
Critical			Environment...	Critical	CallEstablis...	OPERATIO...
Minor			Environment...	Minor	CallEstablis...	OSI_SYSTE...
Warning			Environment...	Warning	AdapterError	OSI_SYSTE...
Warning			Environment...	Warning	CallEstablis...	OSI_SYSTE...
Minor			Environment...	Minor	CallEstablis...	OPERATIO...

DcName	Monitored	Domain Name
dolphin_Admin...	Warning	.dolphin_Admin_Dom
hds.hds_oo1	Warning	.hds.dom1
hds.hds_oo2	Warning	.hds.dom2
demo.radio_oo	Warning	.demo.radio_system

Date	Message
11/14/2000 15:40:32	Disable DC .hds.hds_oo2
11/14/2000 15:40:33	.hds.hds_oo2 successfully disabled
11/14/2000 15:40:35	Enable DC .hds.hds_oo2

OPC Classic – gradniki (nad.)

OPC Classic je opustel/zastarel/zgodovina

OPC-Batch za podporo saržni proizvodnji

- Zgrajen na podlagi IEC 61512-1
- Definira vmesnike za izmenjavo podatkov med napravami
 - Stanja opreme
 - Trenutnih pogojev obratovanja
 - Arhivskih zapisov
 - Vsebine receptov

OPC-DX (ang. Data eXchange) za izmenjavo podatkov

- Standardizirani vmesniki za izmenjavo podatkov med strežniki OPC

OPC-XML (ang. eXtended Markup Language) za izmenjavo podatkov (predhodnik OPC UA)

- Izmenjava podatkov z višjimi sistemi
- Prilagoditev standarda na Microsoftovo usmeritev v .NET
- Uporaba na različnih operacijskih sistemih
- Specificira obliko sporočil za izmenjavo med strežnikom in odjemalcem po protokolih
 - XML (ang. eXtended Markup Language)
 - SOAP (ang. Simple Object Access Protocol)
 - WSDL (ang. Web Service Definition Language)

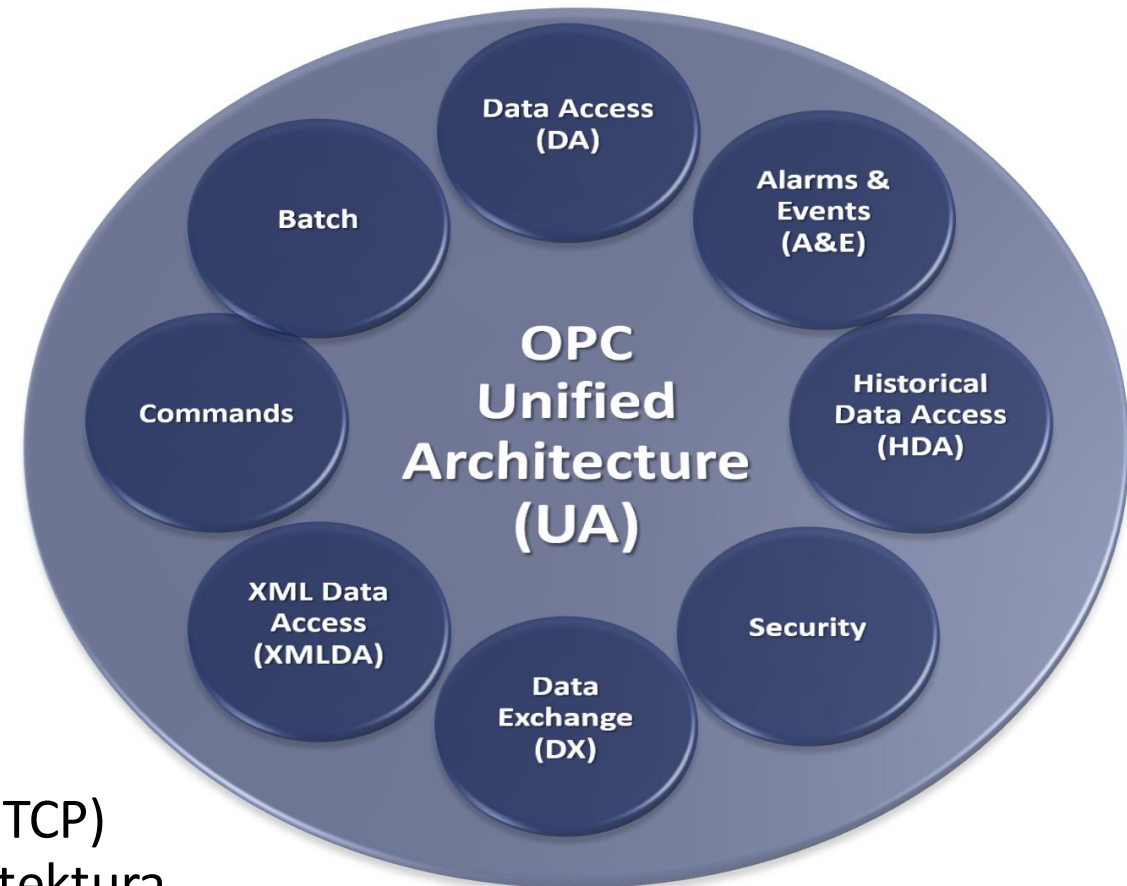
OPC UA

Razširi zelo uporabljan OPC Classic

- Omogoči povezavo PLK in ERP
- Zajemanje podatkov, modeliranje informacij, komunikacija

Uporaba standardnih tehnologij

- Spletne storitve (XML, SOAP, WS)
- Enostavna konfiguracija in vzdrževanje
- Povečana vidnost sistemov (dostop iz več tipov naprav)
- Širše področje uporabe
- Zanesljivost
- Varnost
- Hitrost (OPC binarno kodiranje na protokolu TCP)
- Neodvisnost od platforme (računalniška arhitektura, operacijski sistem, komunikacijski protokol)



OPC UA (nad.)

Raztegljivost

- Profili
 - Neodvisen skupek funkcionalnosti
 - Metode, modeli podatkov, varnost
 - Strežnik sporoča, kaj podpira, odjemalec lahko preveri potrebne funkcionalnosti
 - Primer profila je OPC Data Access

Certifikati

- Certifikati se uporabljajo za varno medsebojni identifikacijo
- Certifikati za strežnik, odjemalec, uporabnika
- Seznam zaupanja vrednih certifikatov na vsaki napravi

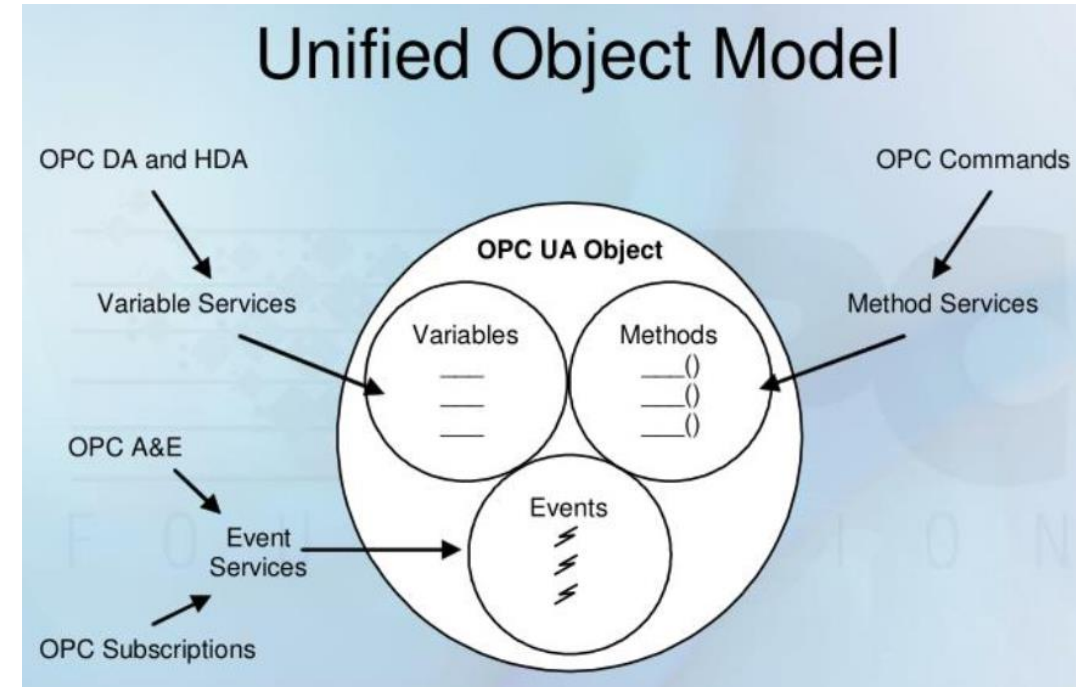
OPC UA (nad.)

Poenotenje vmesnikov

Arhitektura, usmerjena v storitve
(Service Oriented Architecture – SOA)

Enoten nabor storitev

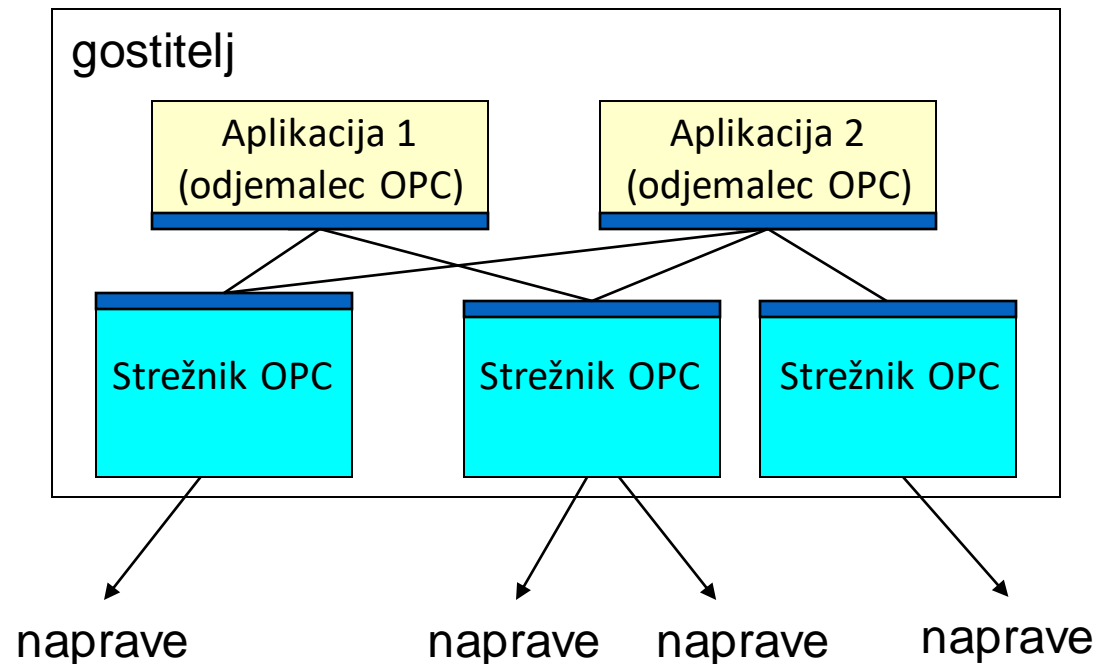
- UA vključuje funkcionalnosti OPC Classic, vendar uporablja samo en nabor storitev
- Storitve
 - poizvedba,
 - branje,
 - pisanje,
 - naročilo, ...
- Poimenovane povezave med vozlišči



Komunikacija

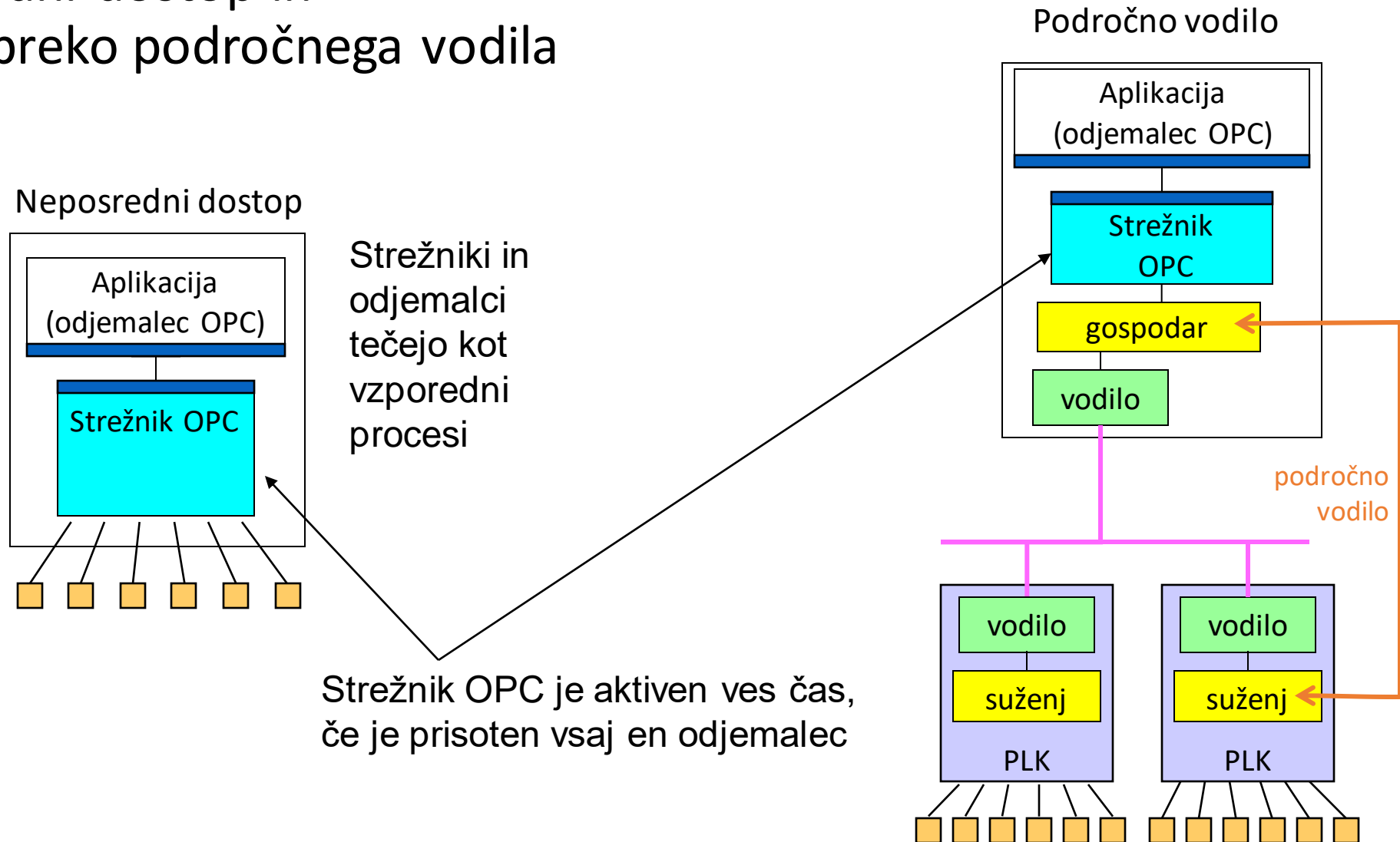
Strežnik OPC in odjemalec OPC na istem računalniku

- Strežniki in odjemalci tečejo kot vzporedni procesi
- Standard OPC podaja vmesnik med strežnikom in odjemalcem v obliki objektov in metod



Komunikacija (nad.)

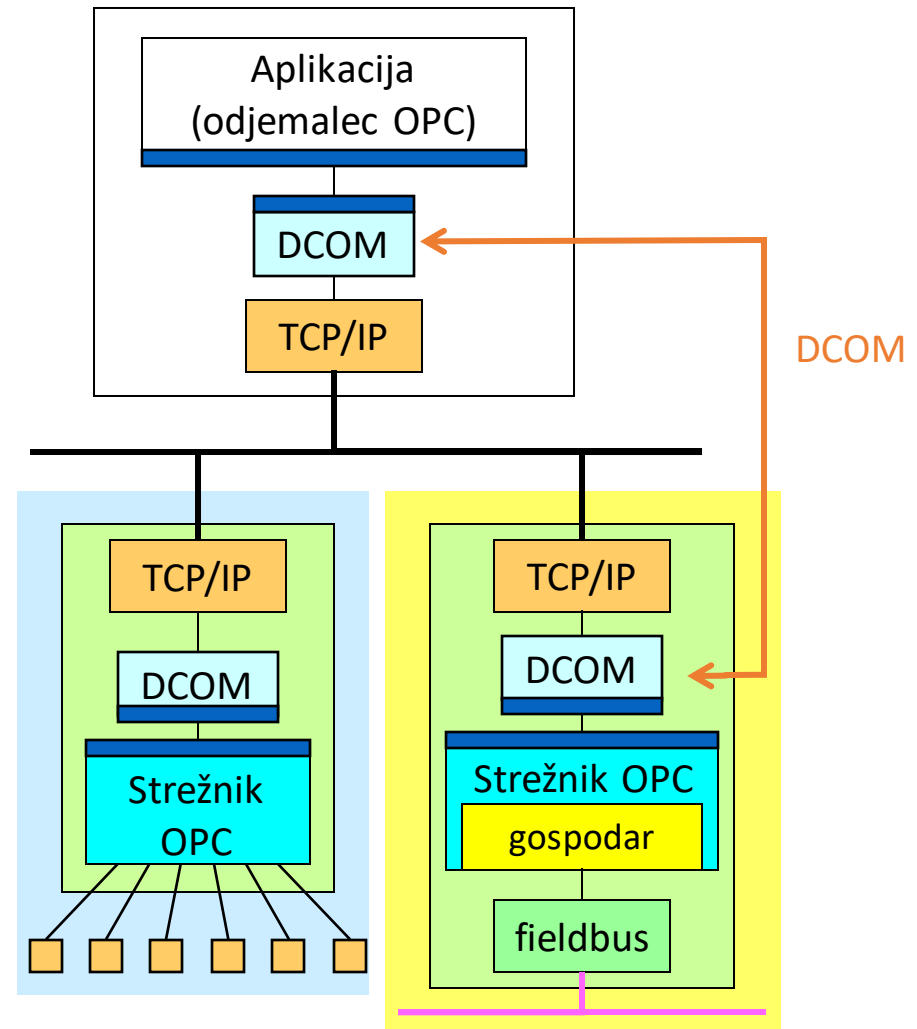
Neposredni dostop in dostop preko področnega vodila



Komunikacija (nad.)

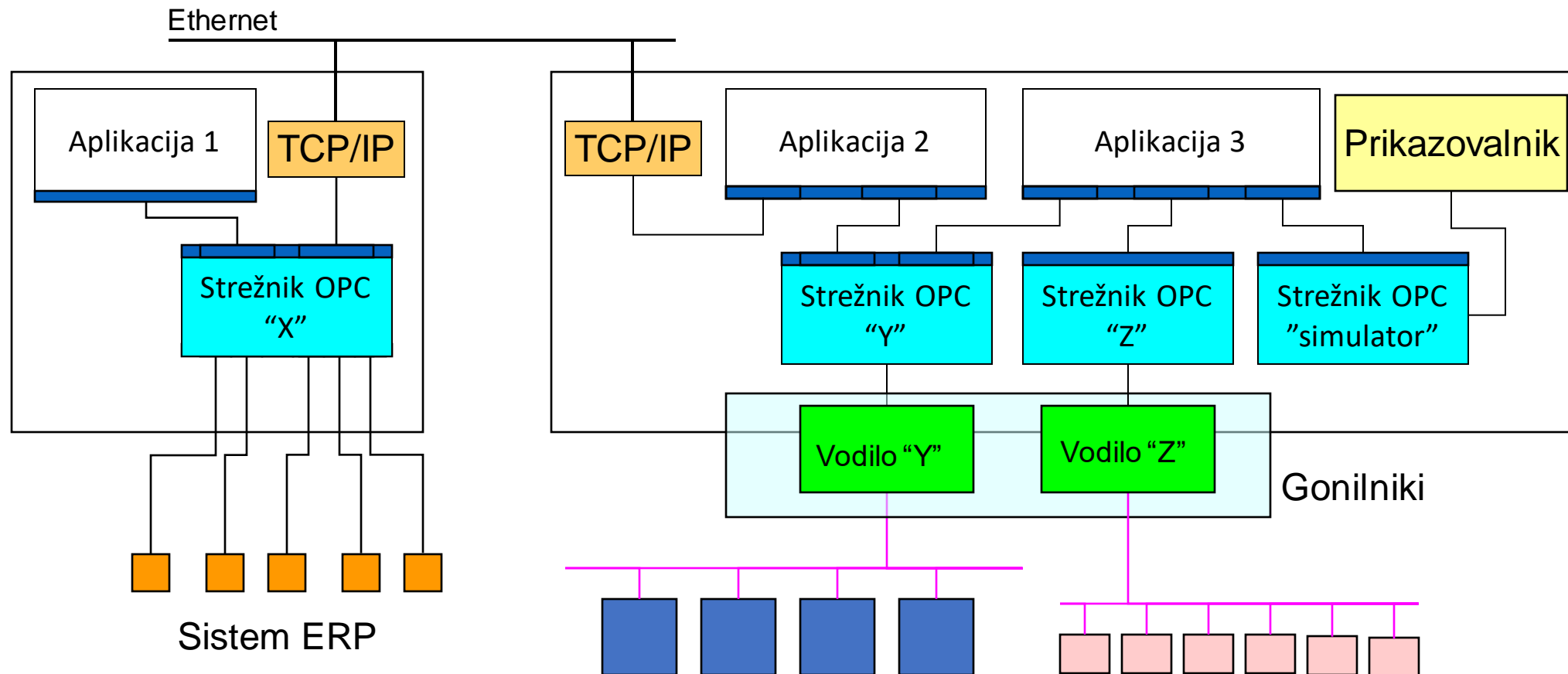
Dostop do strežnika OPC na drugem računalniku

- Problem za **OPC Classic**: požarni zidovi
- Rešitev: OPC UA



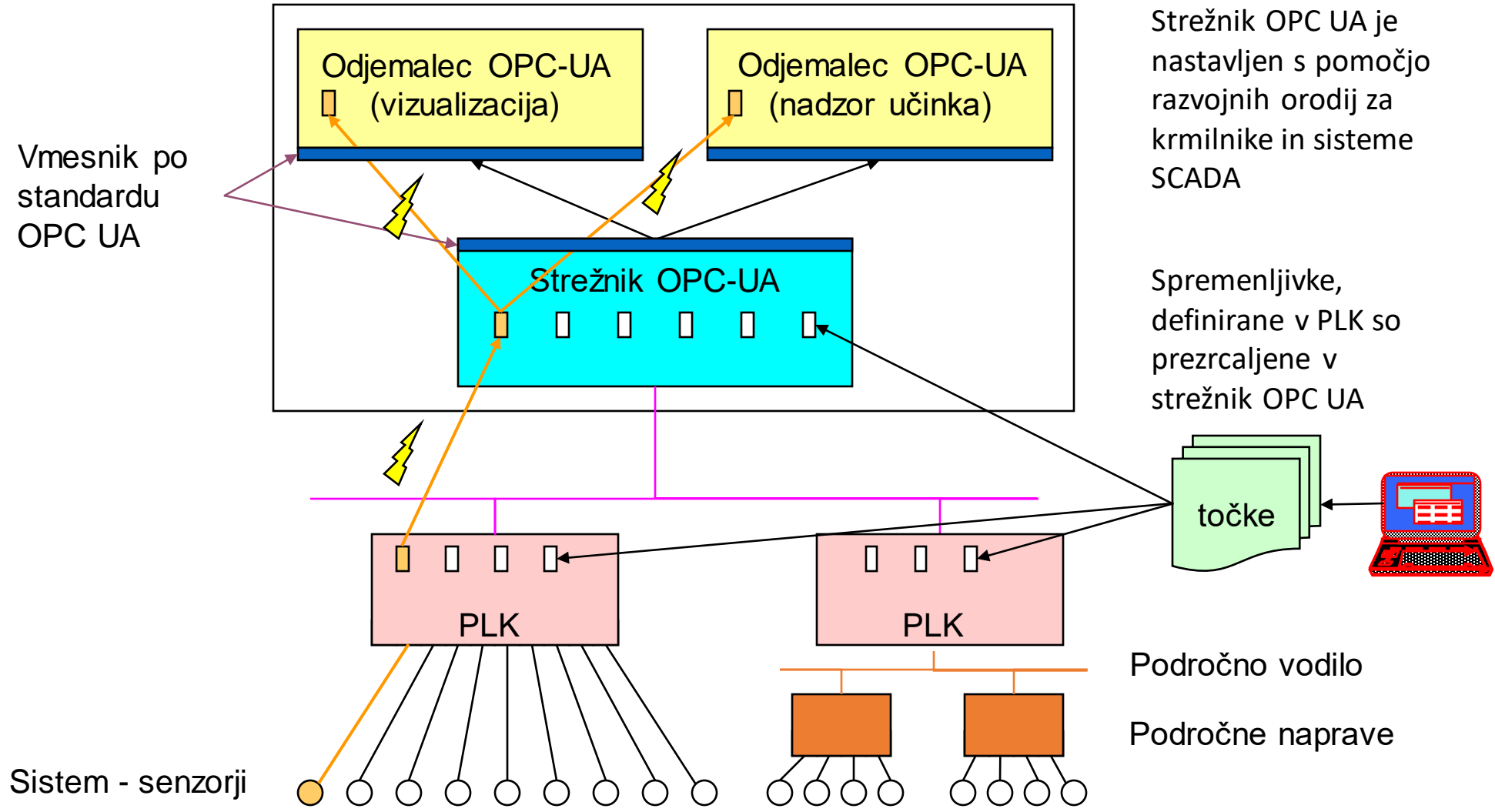
Komunikacija (nad.)

Strežniki OPC podpirajo več odjemalcev na istem ali na različnih računalnikih.



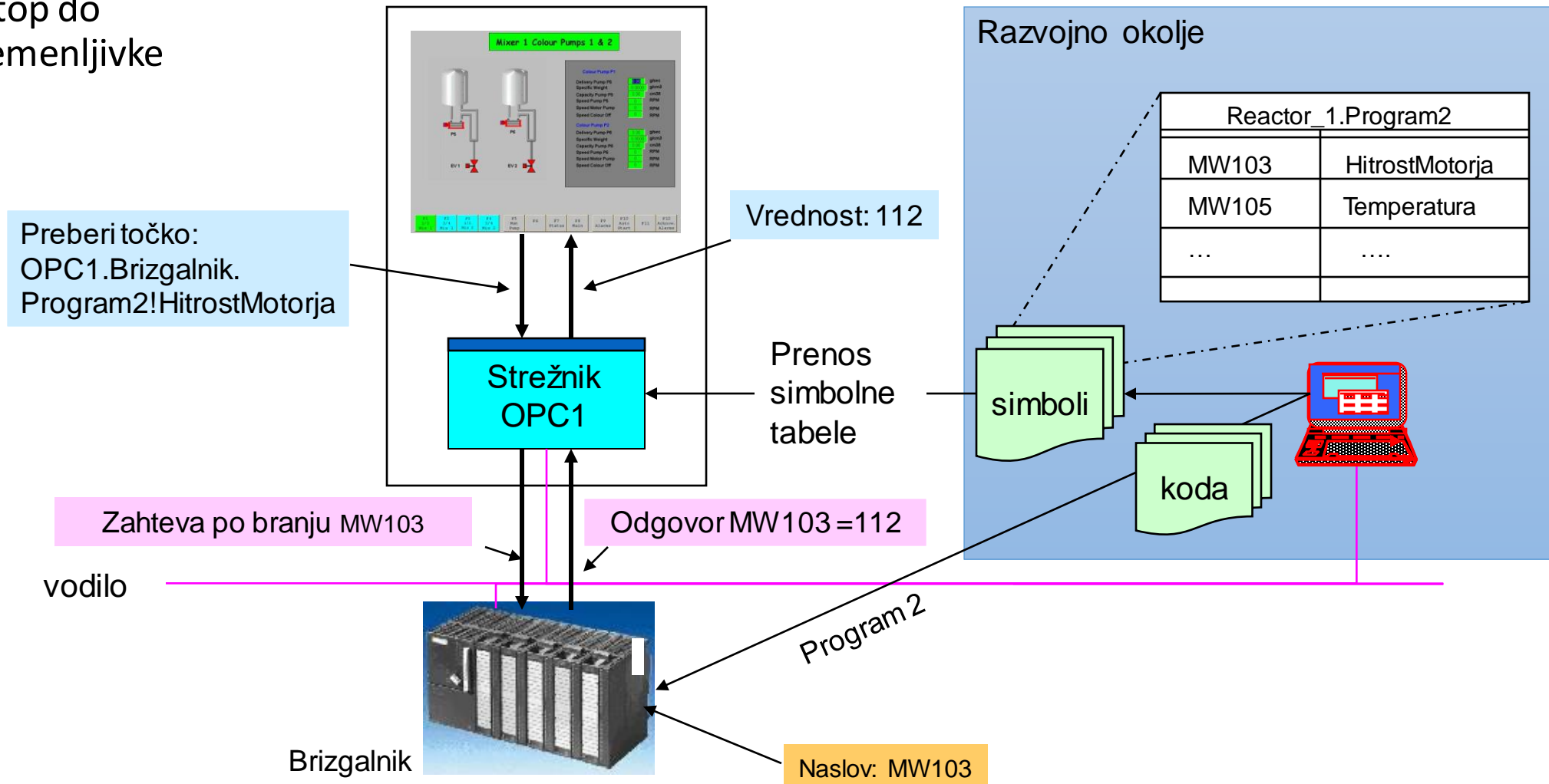
OPC UA: specifikacija Data Access

Shema



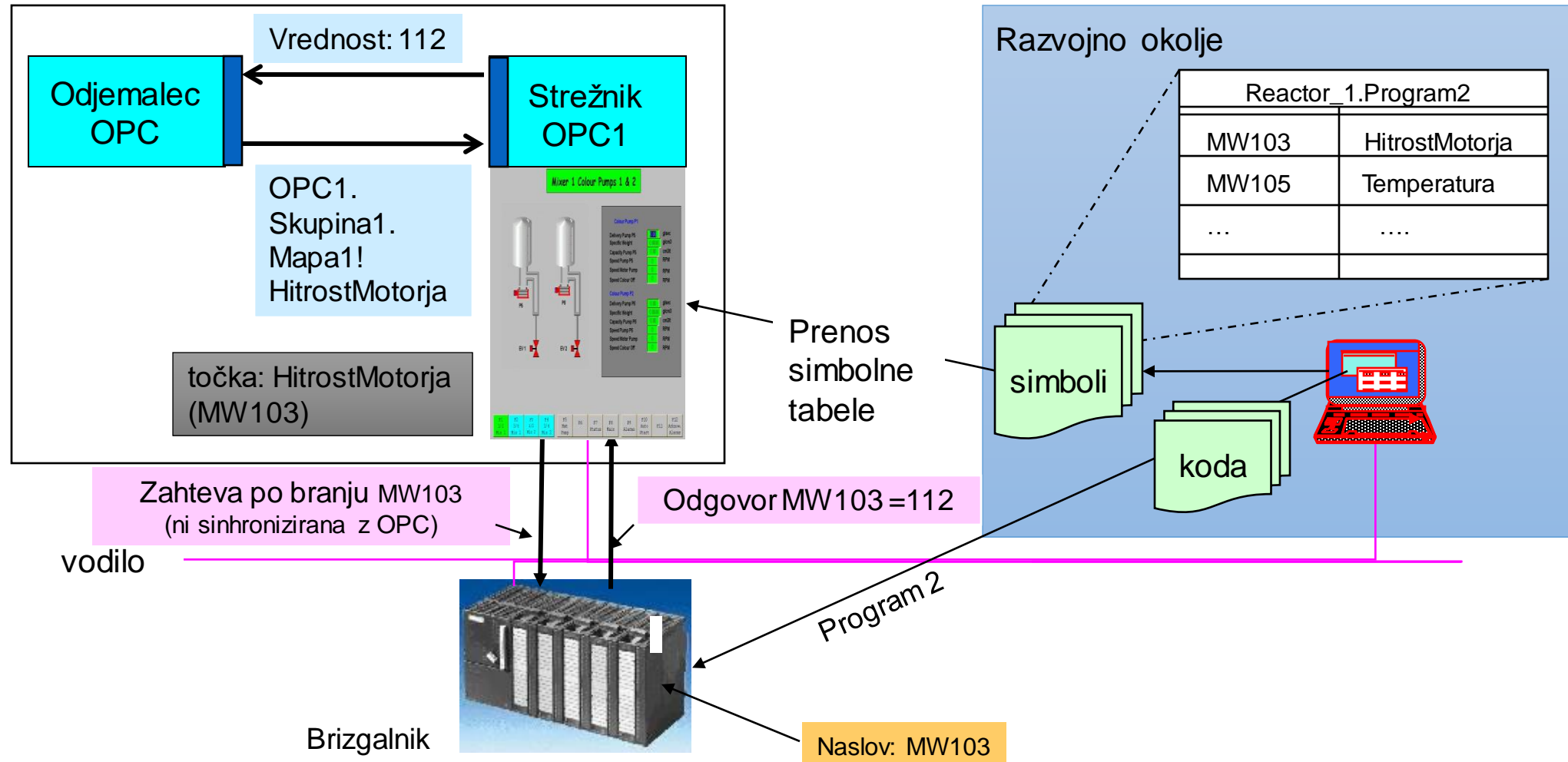
OPC UA: specifikacija Data Access (nad.)

Primer 1:
neposreden
dostop do
spremenljivke



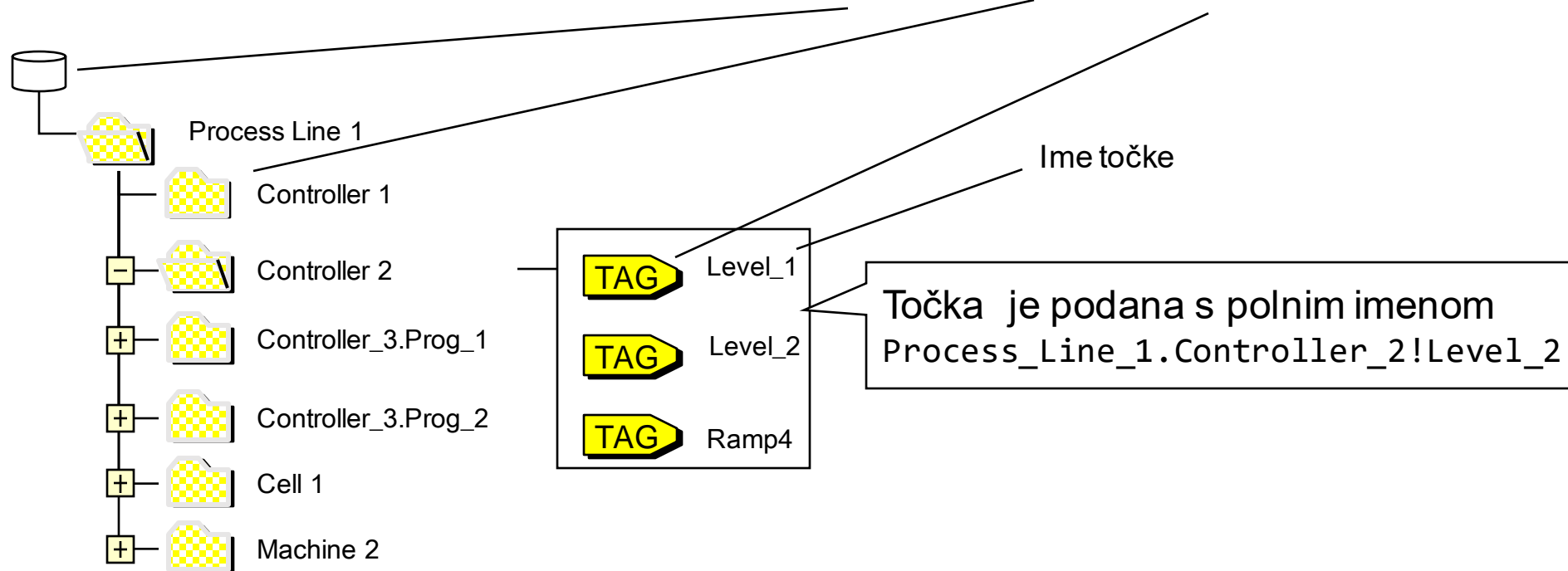
OPC UA: specifikacija Data Access (nad.)

Primer 2: posreden dostop do spremenljivke preko sistema SCADA



Organizacija podatkov

Strežnik OPC je urejen kot mapa s korenom, vejami in listi (točke)



- Veje lahko vsebujejo podveje in točke
- Za strukturo ni nujno, da je hierarhična
- Struktura je vzpostavljena pri konfiguraciji sistema

Organizacija podatkov (nad.)

Podatki so organizirani v vozlišča (ang. nodes), ki sestavljajo drevo

- Celotna množica vozlišč na strežniku tvori naslovni prostor (ang. address space)

Vozlišča so tipizirana

- vsako vozlišče ima attribute, ki so vnaprej določeni glede na tip
- njihove vrednosti se lahko spreminjajo

Vsako vozlišče ima svoj identifikator

- niz, celo število, GUID (ang. globally unique identifier)
- identifikator mora biti unikatni znotraj svojega imenskega prostora (ang. namespace).
- Imenski prostori so določeni s celimi števili.

Najbolj splošna vrsta vozlišč so objekti (ang. objects),

- Lahko vsebujejo spremenljivke (ang. variables)
- Lahko vsebujejo metode (ang. methods).
- Delujejo podobno kot objekti v objektno usmerjenih programskih jezikih.

Operacije na strežniku

Odjemalec vzpostavi povezavo na strežnik

- Strežnik je določen z URL, npr.: `opc.tcp://localhost:4840/`

Odkrivanje strežnikov

Brskanje po strežniku

Branje in spreminjanje spremenljivk

Naročanje na spremembe spremenljivk in druge dogodke

- Ni treba ročno preverjati, kdaj je prišlo do spremembe
- Če ni spremembe, strežnik vseeno to ciklično sporoča
 - Pri tem se ne prenašajo vrednosti
 - Interval je določen na odjemalcu

Klicanje metod

- Objekti imajo lahko pripete metode, ki jih lahko kliče odjemalec

Podatkovni tipi in lastnosti

Vsak element ima definiran podatkovni tip

- Podatkovni tip točke na strežniku dobimo z brskanjem po strežniku (poznamo posebna programska orodja, ki se imenujejo brskalniki).
- Potrebno je poskrbeti, da so podatkovni tipi na strežniku in odjemalcu usklajeni.

```
Boolean,  
Character,  
Byte,           (1 bajt)  
Word,           (2 bajta)  
Double Word,    (4 bajti)  
Short Integer   (2 bajta)  
Integer         (4 bajti)  
Long Integer  
Long Unsigned Integer  
Single Float    (4 bajti)  
Double Float    (8 bajtov)  
Currency  
Date  
String  
Array of ...    (polje)
```

Podatkovni tipi in lastnosti (nad.)

Točk so na odjemalcih predstavljene z naslednjimi dinamičnimi lastnostmi

- Vrednost (ang. Value) prej navedeni podatkovni tipi
- Kvaliteta (ang. Quality) veljavnost zapisa (na primer OK, neustrezen, dvomljiv)
- Časovni pečat (ang. Time Stamp) čas v katerem je bila vrednost prenesena iz PLK na strežnik.
Čas je podan v UTC
(ang. Universal Time Coordinated)

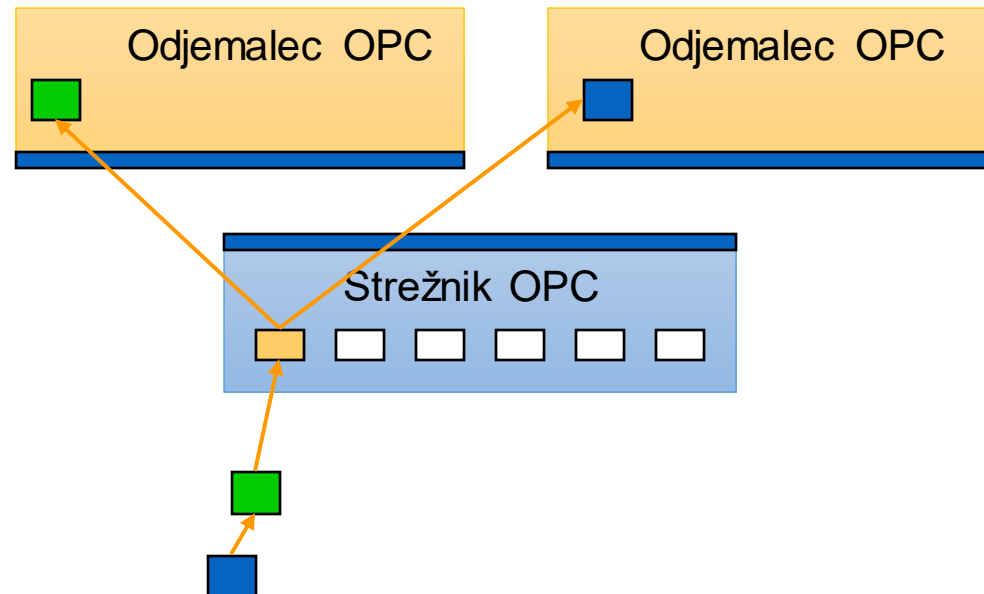
Pri branju se spreminjajo vse tri lastnosti,

Pri pisanju se uporablja samo lastnost *Vrednost*

Programiranje

Branje in pisanje

- Nova vrednost prepiše staro vrednost, ni vrst ali arhivskih zapisov
- Strežnik OPC ne zagotavlja, da različni odjemalci vidijo enako stanje procesa



- Strežnik OPC ne zagotavlja, da so vse spremembe objektov zaznane – v primeru prepočasnega osveževanja lahko kakšne zgrešimo

Programiranje (nad.)

Branje ob dogodkih

- Funkcija se kliče vsakič, ko pride do spremembe elementa v skupini
- Spremenljivka je določena z naslovom na odjemalcu, ne s polnim imenom
- Za vsako točko dobimo vrednost, kvaliteto in časovni zapis

- Odjemalec OPC lahko pokliče funkcijo tudi v primeru, da se spremeni vrednost v katerikoli od skupin (ang. Global Data Change)
- Aplikacija mora ugotoviti, v katero skupino in h kateri točki spremembe sodijo

Programiranje (nad.)

Knjižnice

- Za zagotovitev sledenja standardu OPC Foundation na svoji spletni strani ponuja knjižnice, potrebne za dostop do strežnikov OPC
 - koda v C in Javi ni več vzdrževana
 - .NET Core: <https://github.com/OPCFoundation/UA-.NETStandard>
- Proizvajalci strežnikov ne implementirajo vedno vseh funkcij, zato se lahko zgodi, da pride do nepričakovanih napak

Beckhoff TwinCAT

- [Infosys: TF6100](#)
- [Predstavitev in dobre prakse](#)

Orodja, povezave

OPC Foundation

- <https://opcfoundation.org/>
- specifikacije standarda (samo za člane)
- strežniki, odjemalci, brskalniki, simulatorji

Python (FreeOpcUa)

- python-opcua (zastarel): <https://github.com/FreeOpcUa/python-opcua>
- asinhroni API: <https://github.com/FreeOpcUa/opcua-asyncio>
- odjemalec z grafičnim vmesnikom: <https://github.com/FreeOpcUa/opcua-client-gui>

NodeJS

<https://node-opcua.github.io/>

UA Expert: polno funkcionalni odjemalec/brskalnik

<https://www.unified-automation.com/products/development-tools/uaexpert.html>