

Szabvány web szerver specifikáció alkalmazási lehetőségei

- Bemutató
- Kihívásaink
- Web szerver
- YAML
- Web szerver vs. LabVIEW
- Megvalósítás
- Felhasználás



Bánhegyi Mihály

- Senior Software Developer



- 2004 óta LabVIEW
- KEAT Architect



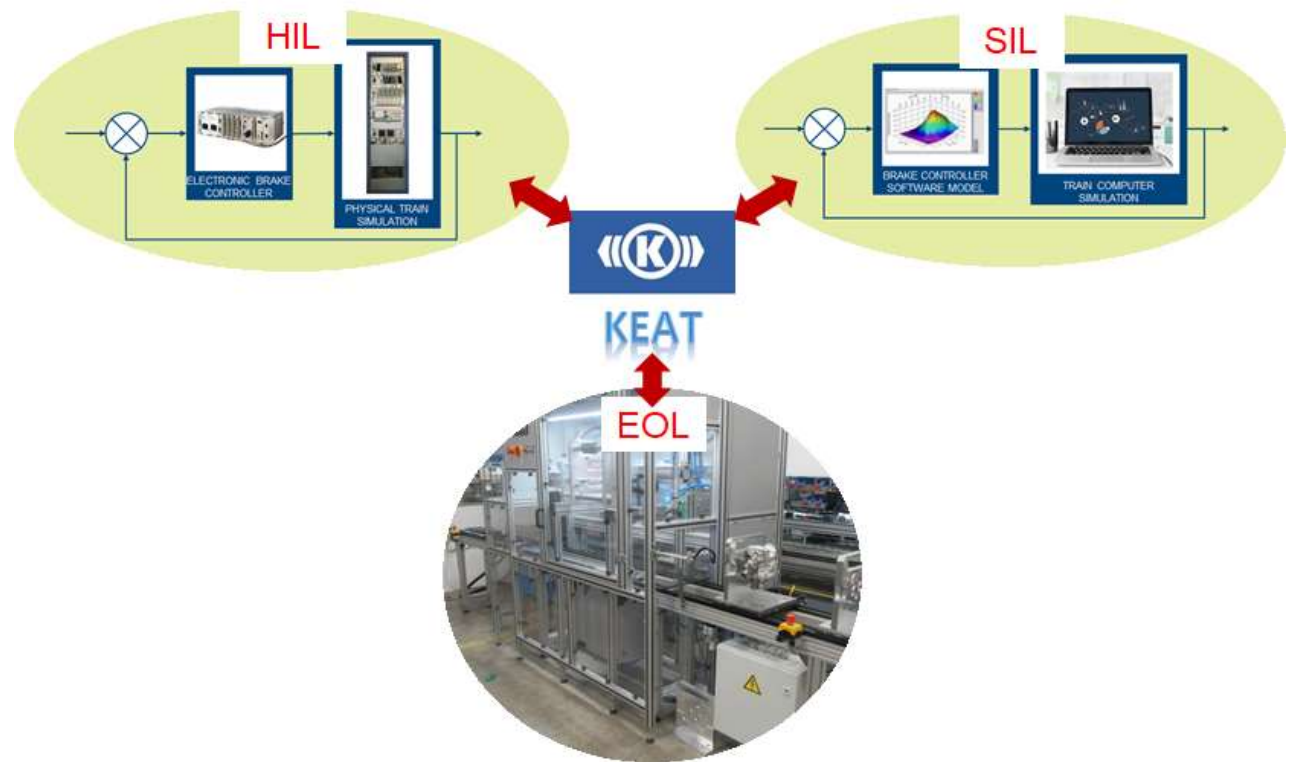
Kihívások – 1.

- Vasúti alkalmazások – vasúti követelmények
 - 30 év élekciklus termékre és tesztkörnyezetre egyaránt
 - EN50128 szerinti dokumentáció
- LabVIEW – C, C++, C# – Python – Oracle technológiák integrálása egyetlen platformban



Kihívások – 1.

- >20k egyedi teszt projekt vs. 1 teszt platform
- IO pontok száma: 300 – 50,000
- Tesztlépések száma: 500 – 4,000,000



Mi a web szerver?

- A web szerver egy software és hardware amelyik a kliensek kéréseire HTTP és egyéb protokollt használva válaszol. A web szerver fő feladata honlap tartalmakat megjeleníteni web oldal tároláson, feldolgozáson és szállításon keresztül a felhasználók számára.



HTTP

- HyperText Transfer Protocol
 - TCPIP alapú
 - Fő részei:
 - URL
 - Header
 - Body
 - Metódusok:
 - HEAD
 - GET
 - POST
 - PUT
 - DELETE
 - OPTIONS
 - TRACE, CONNECT, UPGRADE, PATCH..



HTTPS

- SSL / TLS titkosító / autentikáló réteg a HTTP és TCPIP réteg között



Mire jó a web szerver?

- Megbízható, skálázható és biztonságos kapcsolat
- OS és nyelv független interfész
- Lehetőség REST architektúra kialakítására



REST

- REpresentational State Transfer
- Jellemzők:
 - Szerver és Kliensek elkülönítése
 - Egységes interfész
 - Állapotmentes
 - Réteges felépítés
 - Gyorsítótárazható



Mi a YAML?

- Yet Another Markup Language
- Jellemzők:
 - Emberi olvashatóság
 - Kulcs-érték párok
 - Minimális szintaktika
 - Adat szerializálási célok



Mi a YAML?

- Yet Another Markup Language
- Jellemzők:
 - Emberi olvashatóság
 - Kulcs-érték párok
 - Minimális szintaktika
 - Adat serializálási célok



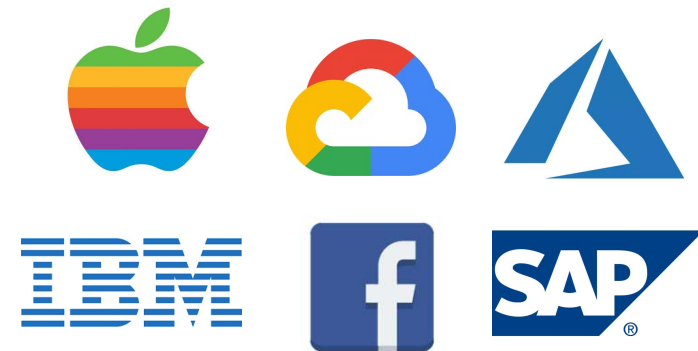
Mi a YAML?

- Yet Another Markup Language
- Jellemzők:
 - Emberi olvashatóság
 - Kulcs-érték párok
 - Minimális szintaktika
 - Adat serializálási célok
 - OpenAPI specifikáció formátuma



Mi a YAML?

- Yet Another Markup Language
- Jellemzők:
 - Emberi olvashatóság
 - Kulcs-érték párok
 - Minimális szintaktika
 - Adat serializálási célok
 - OpenAPI specifikáció formátuma
- [Swagger Editor](#)
- [Swagger UI](#)

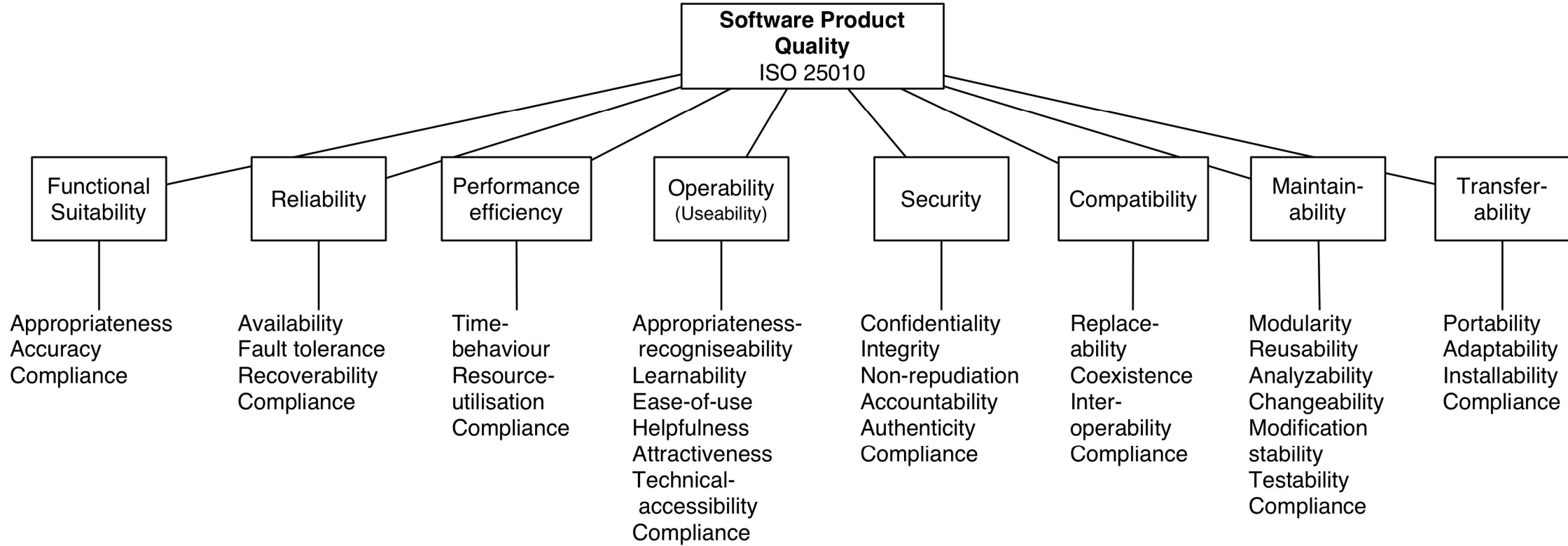


Kihívások – 2.

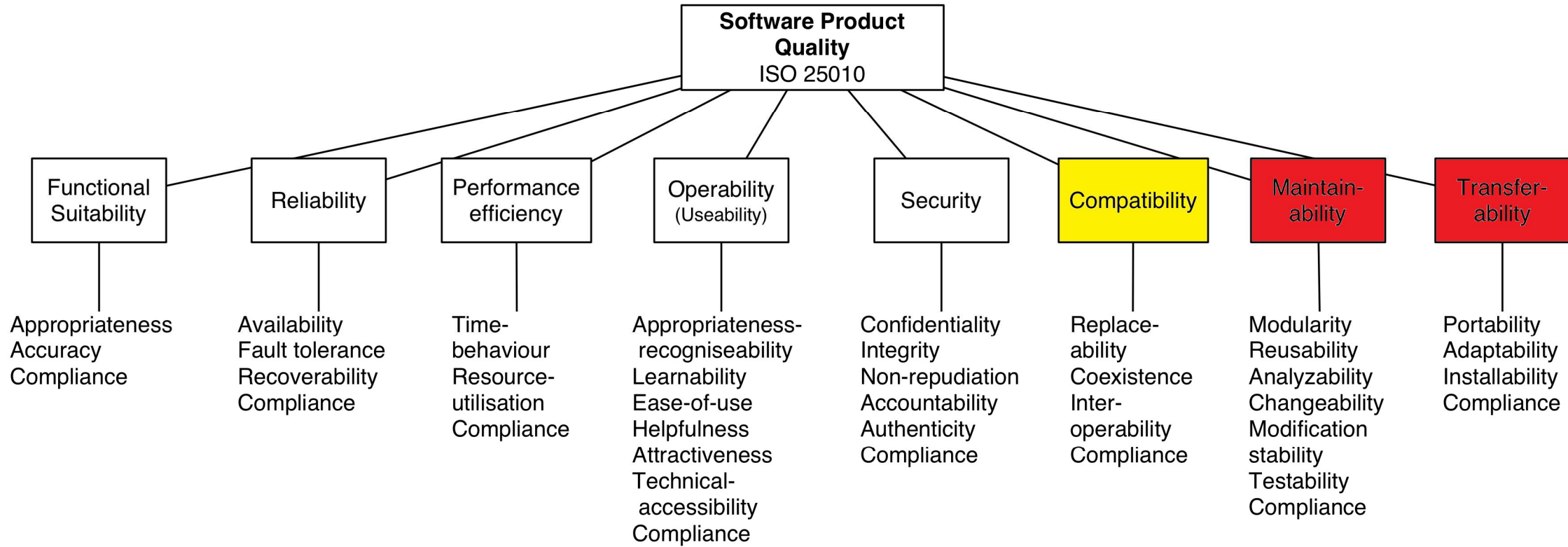
- LabVIEW web szerver projekthez kötött
- Limitált URL útvonal hossz és tartalom lehetőségek
- Limitált HTTP/HTTPS metódusok
- Limitált szerver certifikációs lehetőségek



Szoftver minőség vs. LabVIEW projekthez kötött megoldások



Szoftver minőség vs. LabVIEW projekthez kötött megoldások



Célok

- Ugyanazon web szerver interfész használata egymástól eltérő LabVIEW projektben
- Komplex URL útvonalak használata
- LabVIEW által alapértelmezetten nem támogatott HTTP/HTTPS metódusok használata
- Szabadon, projekt függetlenül konfigurálható szerver certifikációs lehetőségek



Megvalósítás

- Portolható web szerver modul
- Automatikus kódgenerálás

Portolható web szerver modul

- Natív TCPIP csomag kezelés

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`

- URL séma
- Hostname vagy tartománynév : port szám opcionális (HTTP: 80, HTTPS: 443)
- UriTag = Osztályok
- PARAM: kötelező UriTag paraméter értékek kötött sorrendben
- param=value: opcionális UriTag paraméter név/érték párok bármilyen sorrendben

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`

- URL séma
- Hostname vagy tartománynév : port szám opcionális (HTTP: 80, HTTPS: 443)
- UriTag = Osztályok
- PARAM: kötelező UriTag paraméter értékek kötött sorrendben
- param=value: opcionális UriTag paraméter név/érték párok bármilyen sorrendben

- UriTag paraméterek függenek az UriTag kontextusától
- Pl.: `http://hostname/Version?ApplicationName=ExampleApp`
`http://hostname/Version/ExampleApp/Version?PluginName=ExamplePlugin`

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`

- URL séma
- Hostname vagy tartománynév : port szám opcionális (HTTP: 80, HTTPS: 443)
- UriTag = Osztályok
- PARAM: kötelező UriTag paraméter értékek kötött sorrendben
- param=value: opcionális UriTag paraméter név/érték párok bármilyen sorrendben

- UriTag paraméterek függenek az UriTag kontextusától
- Pl.: `http://hostname/Version?ApplicationName=ExampleApp`
`http://hostname/Version/ExampleApp/Version?PluginName=ExamplePlugin`
- UriTag osztály konstans: leíró a paraméterek kontextusáról

Portolható web szerver modul

- Natív TCPIP csomag kezelés
- HTTPS Alapfeltétel: natív TLS támogatás (LabVIEW2020)
- `http://hostname:port/UriTag1/{PARAM}/{PARAM}/UriTag2/..UriTagX?param=value¶m=value`
- HTTP metódusok: GET, POST, PUT, DELETE...

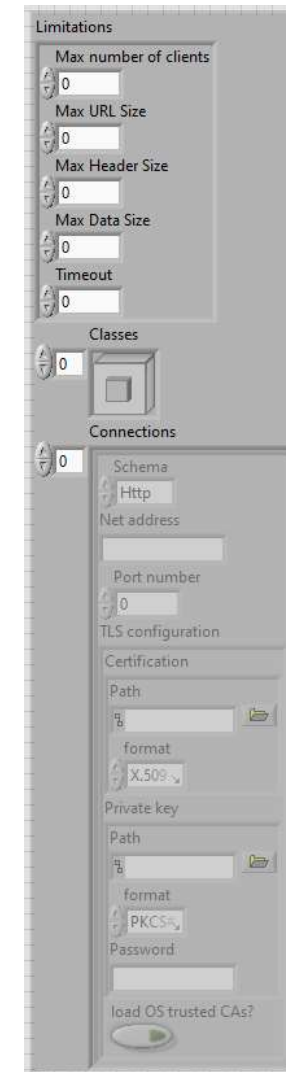
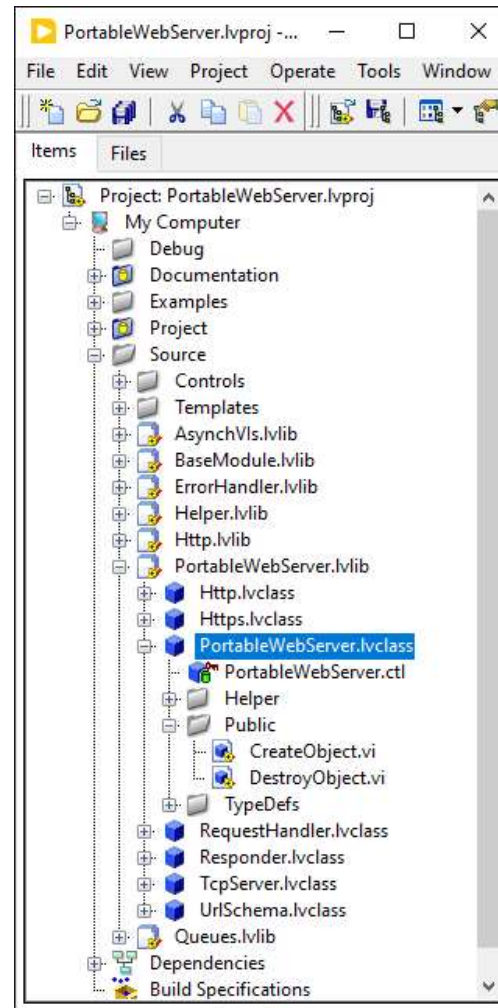
- **URL séma**
- **Hostname vagy tartománynév** : port szám opcionális (HTTP: 80, HTTPS: 443)
- **UriTag** = Osztályok
- **PARAM**: kötelező **UriTag** paraméter értékek kötött sorrendben
- **param=value**: opcionális **UriTag** paraméter név/érték párok bármilyen sorrendben

- **UriTag** paraméterek függenek az **UriTag** kontextusától
- Pl.: `http://hostname/Version?ApplicationName=ExampleApp`
`http://hostname/Version/ExampleApp/Version?PluginName=ExamplePlugin`
- **UriTag** osztály konstans: leíró a paraméterek kontextusáról

- HTTP metódus=**UriTag** függvénye – dynamic dispatch template

Portolható web szerver modul

- Gateway
- Konfiguráció:
 - Határértékek
 - **UriTag** osztályok: regisztráció
 - Kapcsolatok: HTTP és/vagy HTTPS párhuzamosan

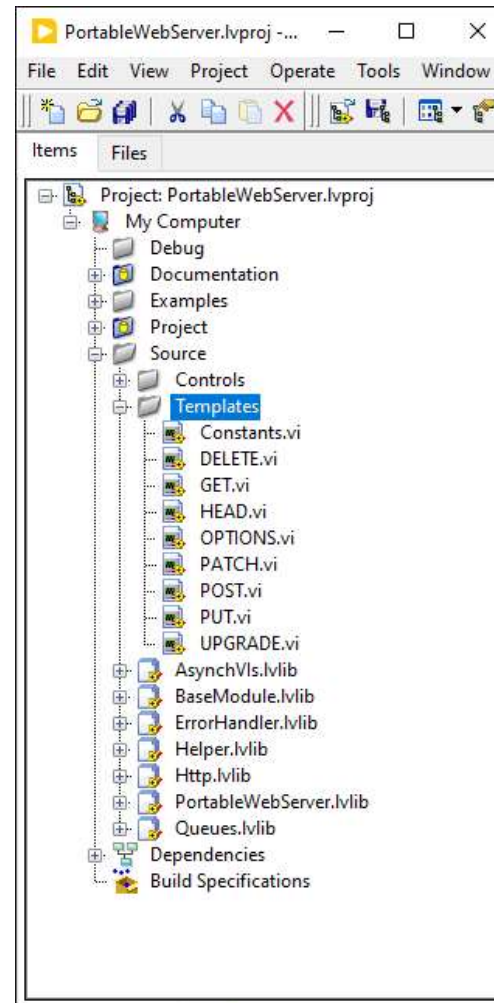


Portolható web szerver modul

- **UriTag** osztályok regisztrációja:
 - Nincs függőségi viszony az **UriTag** osztályok között
 - **UriTag** osztályok gyerek osztályai is beregisztrálhatnak

Portolható web szerver modul

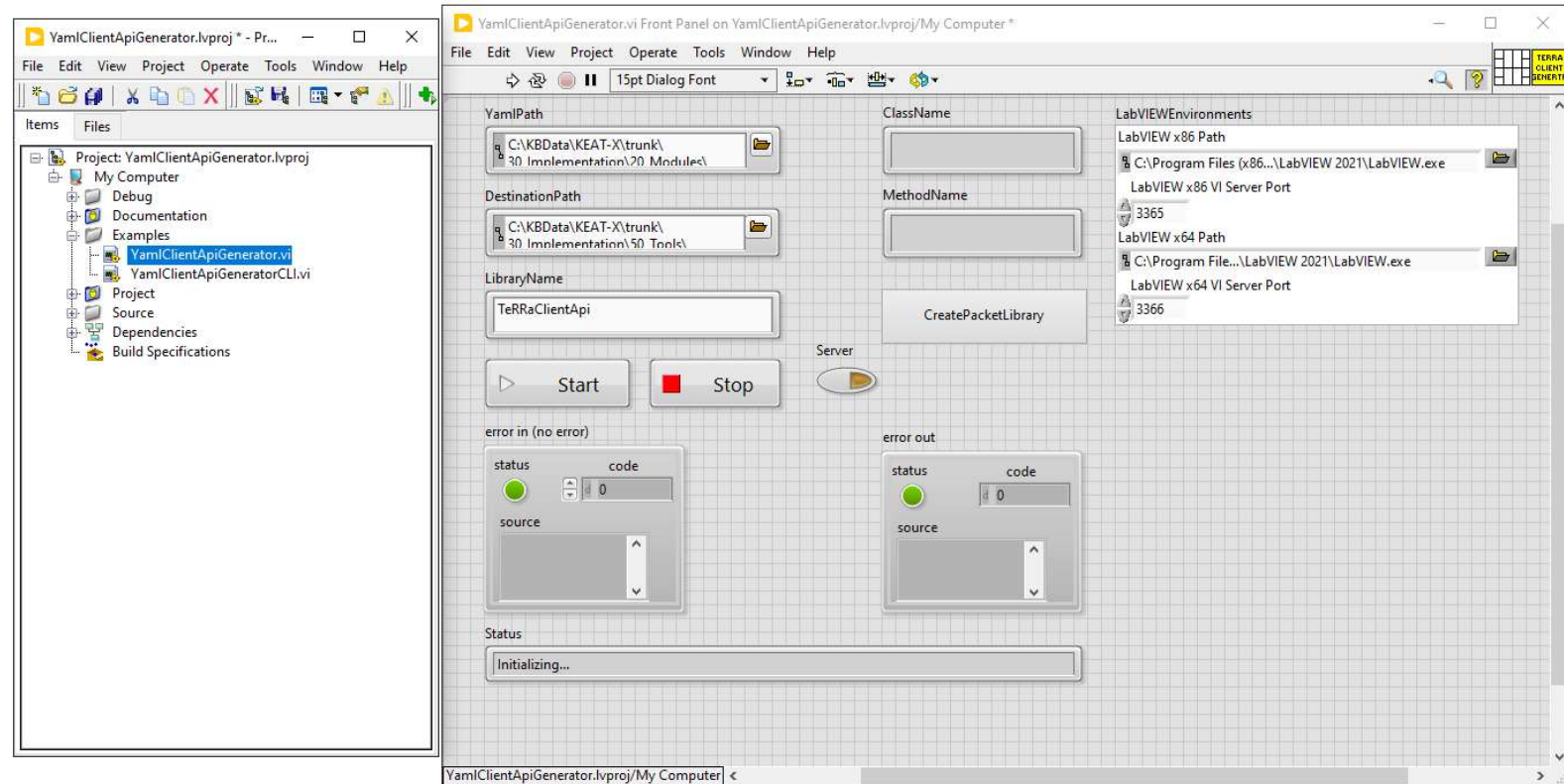
- Mintafüggvények az **UrlTag** osztályokhoz
 - "Constants.vi" – kötelező elem **paraméterek** kontextusához
 - HTTP metódusok - opcionális



Élő demó

Automatikus kódgenerálás

- YAML: kulcs/érték párok = Variant LUT
- VI Scripting:
 - Kliens
 - Szerver
- Continuous Integration

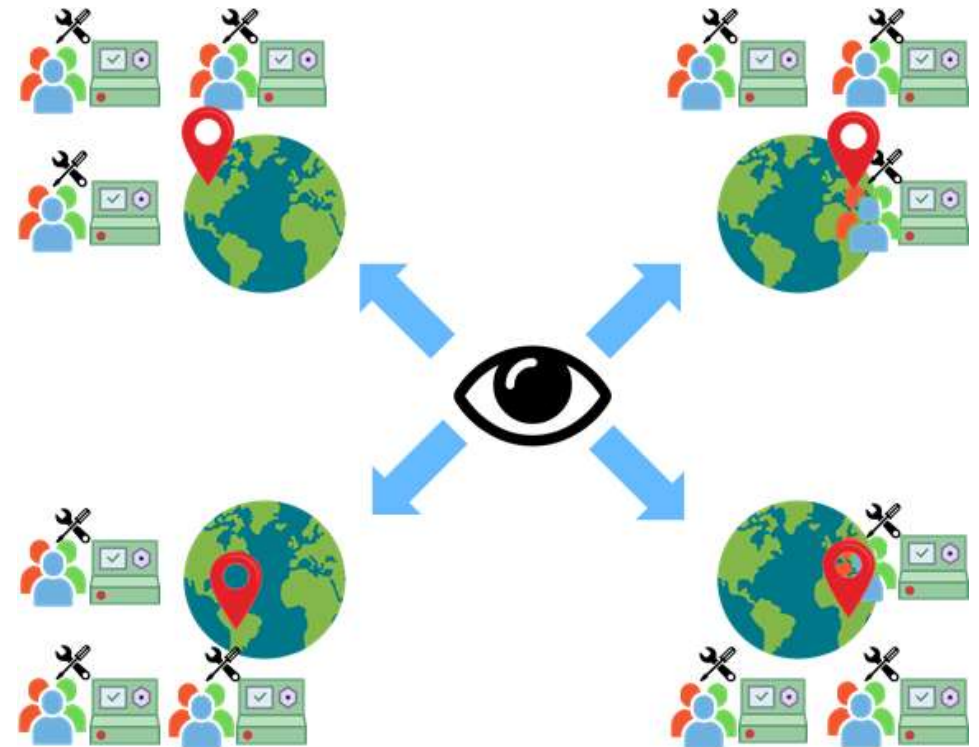
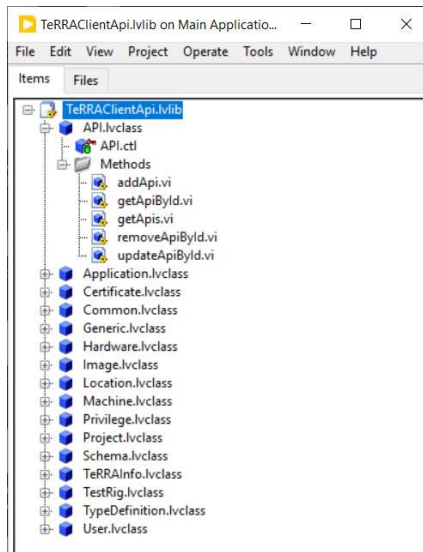


Felhasználás

- Teszt rendszer felügyelet
- Távoli alkalmazás felügyelet
- Modell generálás
- Headless tesztautomatizálás

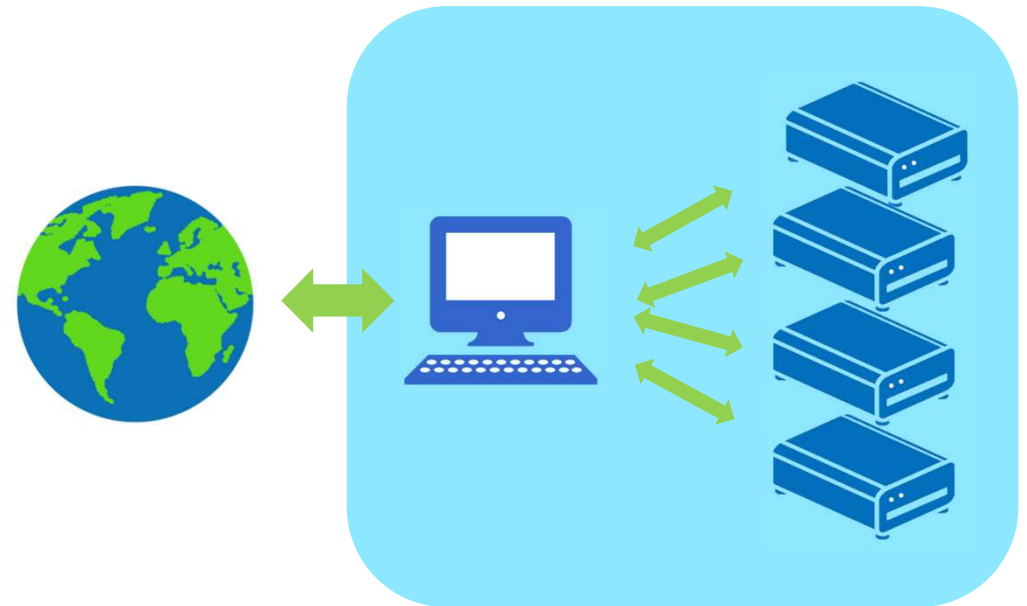
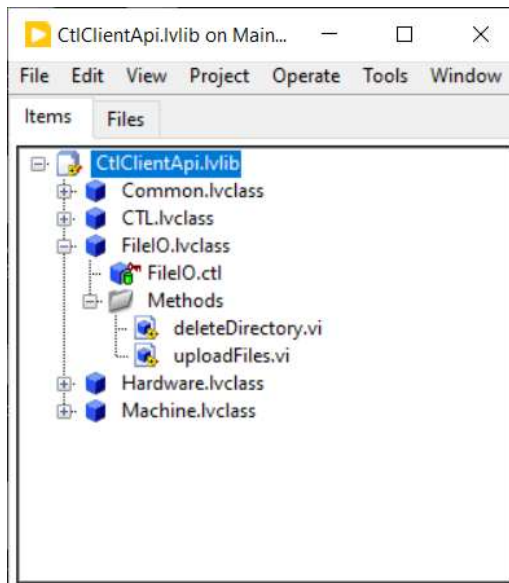
Teszt rendszer felügyelet - kliens

- Szerver: Oracle adatbázis Apache web server interfésszel
- Teszt rendszer konfigurálás
- Teszt rendszer monitorozás
- Felhasználói jogosultság kezelés
- Teszt projekt konfigurálás
- ...



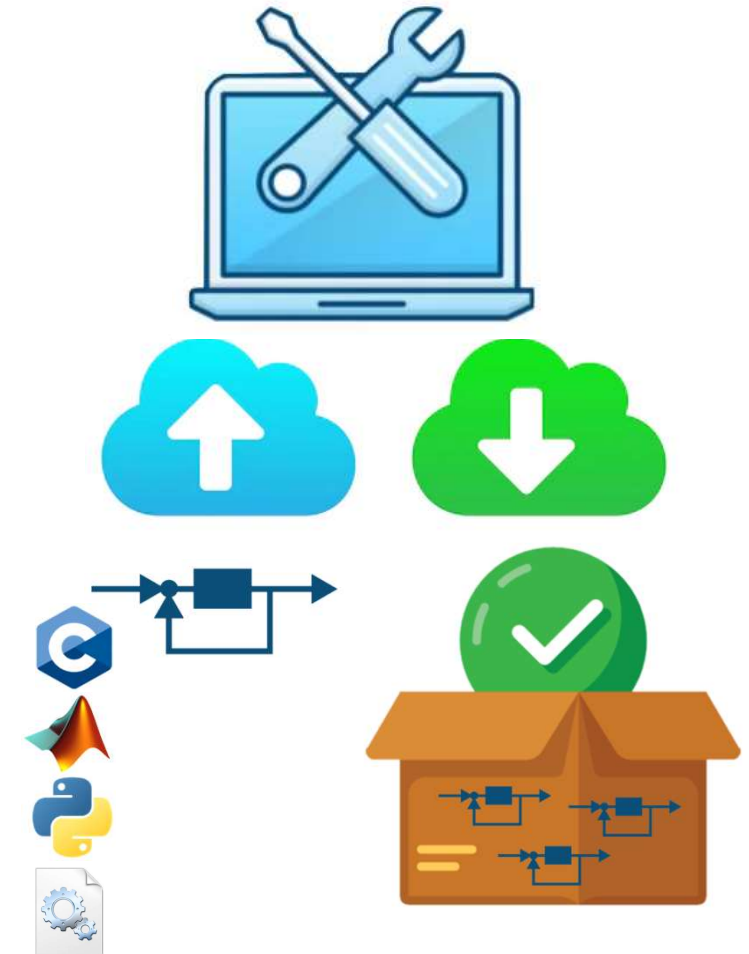
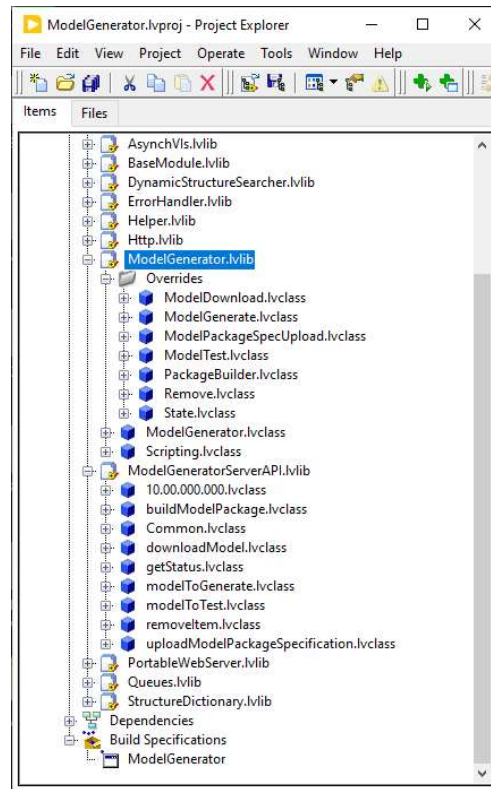
Távoli alkalmazás felügyelet - kliens

- Szerver: Python szerviz alkalmazás
- Alkalmazás letöltés, indítás, leállítás
- Gép monitorozás
- Hardver monitorozás



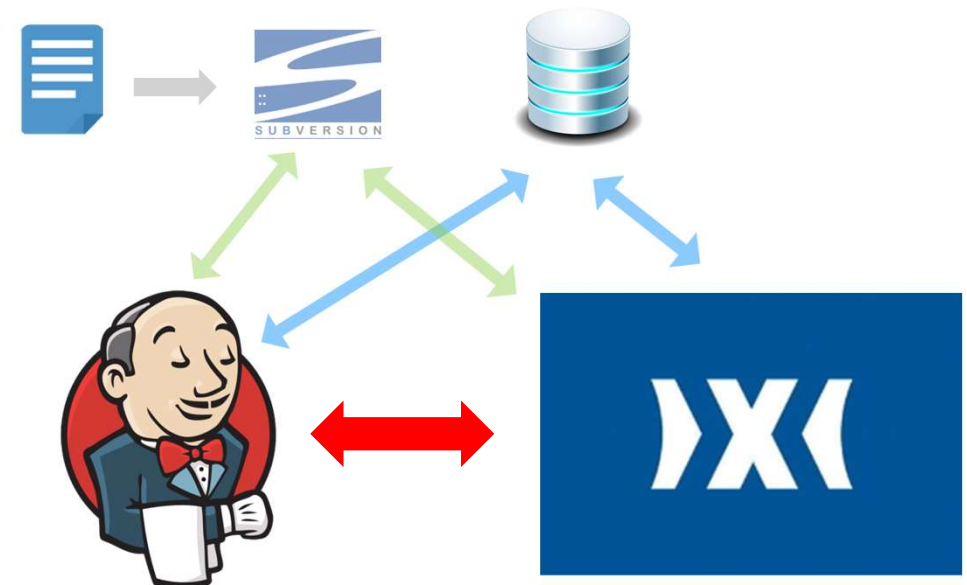
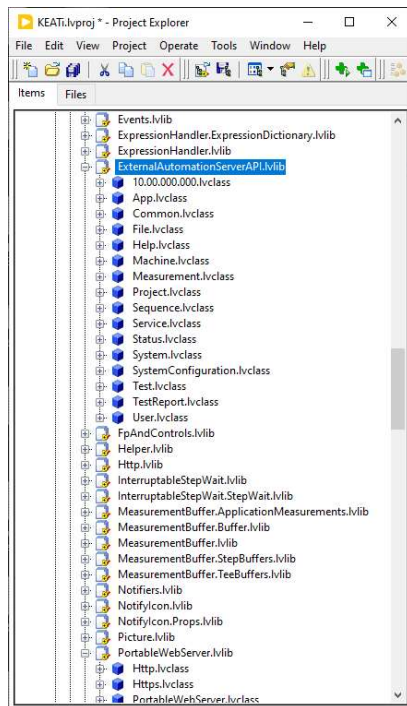
Modell generálás - szerver

- Szerver: LabVIEW alkalmazás, Kliensek: C# alkalmazások
- Felhasználói modellek (C, Matlab, Python, LabVIEW..) feltöltése, ellenőrzése, keretrendszerbe csomagolása, tesztelése, letöltése



Headless tesztautomatizálás - szerver

- Szerver: LabVIEW alkalmazás, Kliens: Jenkins Job – Python script
- Fékvezérlő szoftver automatikus tesztelése



Kérdések?

Köszönöm a figyelmet!