

Az MTA Cloud a tudományos alkalmazások támogatására



Kacsuk Péter
MTA SZTAKI

Kacsuk.Peter@sztaki.mta.hu

- Kétféle skálázhatóság:
 - **Vertikális**: dinamikusan változik a VM mérete (CPU core száma, memória, tároló) – általában nem támogatott
 - **Horizontális**: a virtuális gépek száma dinamikusan változik – ezt lehet különböző eszközökkel támogatni, ez **egyfajta párhuzamos végrehajtást feltételez a felhőben**



Erről szól ez az előadás

Egyszerű (1 lépésből álló) alkalmazások:

1. Egyszerű szekvenciális algoritmus végrehajtása

1. párhuzamosítás nélkül



2. HTC (high throughput computing) párhuzamosítással. Tipikus elnevezések:

1. lazán csatolt párhuzamosság

2. “Parameter sweep” párhuzamosság

3. “Embarrassly parallel” párhuzamosság



2. HPC (high performance computing) algoritmus végrehajtása

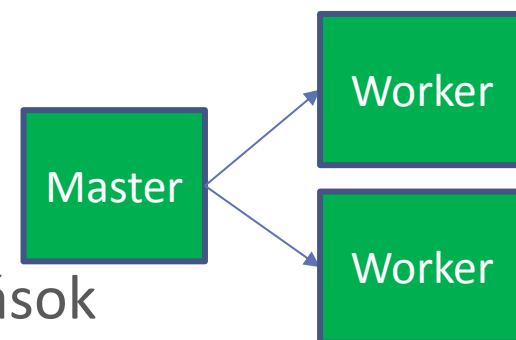
1. MPI alkalmazás

2. OpenMP alkalmazás



Összetett (több lépésből álló) alkalmazások:

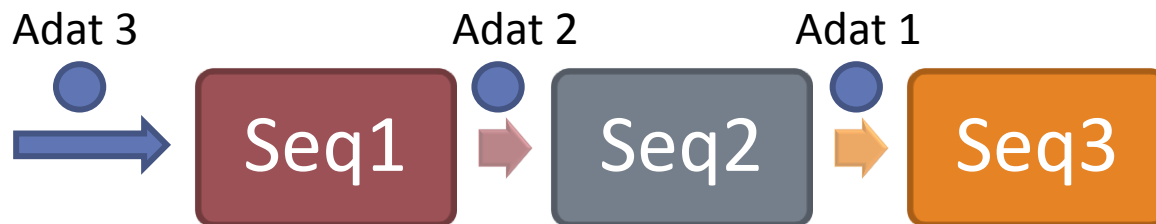
1. Docker alkalmazás végrehajtása
 1. Párhuzamosítás nélkül
 2. Párhuzamosítással docker klaszteren
2. PBS (Condor, stb.) klaszteren futó alkalmazások
3. Mapreduce alkalmazás végrehajtása Hadoop klaszteren
4. Pipeline alkalmazás végrehajtása
 1. Egyedi adattal működő pipeline
 2. Adat folyam (data stream) pipeline
5. Workflow alkalmazás (a fentiek kombinációja) végrehajtása



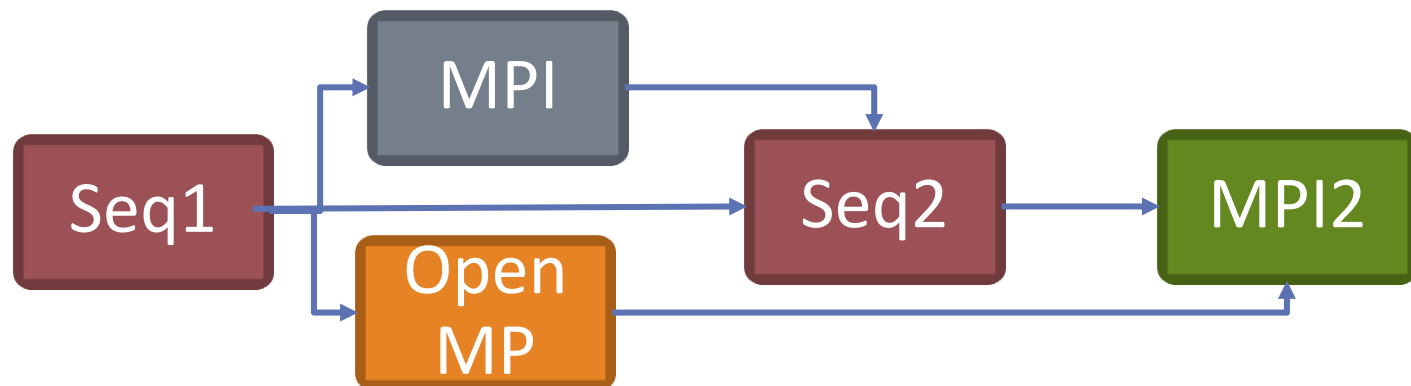
5.1 Egyedi pipeline:



5.2 Adat folyam pipeline:



6. workflow alkalmazás:



1.1 Egyszerű szekvenciális algoritmus végrehajtása párhuzamosítás nélkül

- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő Linux vagy Windows gép gyors létrehozása a felhőben
 - Linux és Windows lemezképeket adunk, amikből a kívánt gépek indíthatóak
 - Négyféle mérettel lehet ezeket elindítani (ld. Következő slide)
 - További kiegészítő tároló (volume) csatlakoztatása a VM-hez
 - Feldolgozandó adatok feltöltése a csatolt tárolóba
 2. Szekvenciális alkalmazás telepítése a létrehozott gépen
 3. A létrehozott gépről elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen

Az Openstack 5-féle VM alaptípust támogat

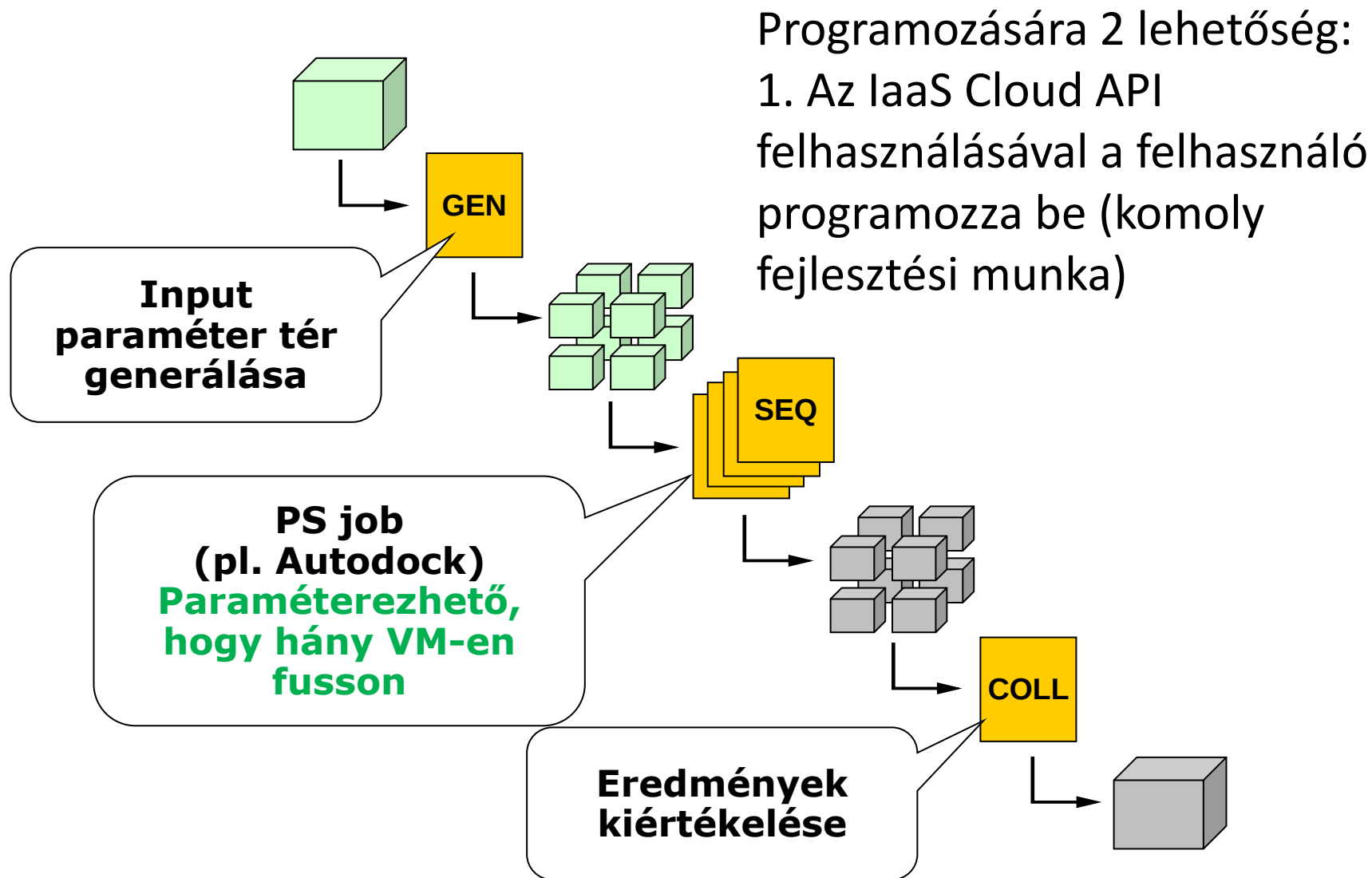


```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs
1	m1.tiny	512	1	0		1
2	m1.small	2048	20	0		1
3	m1.medium	4096	40	0		2
4	m1.large	8192	80	0		4
5	m1.xlarge	16384	160	0		8

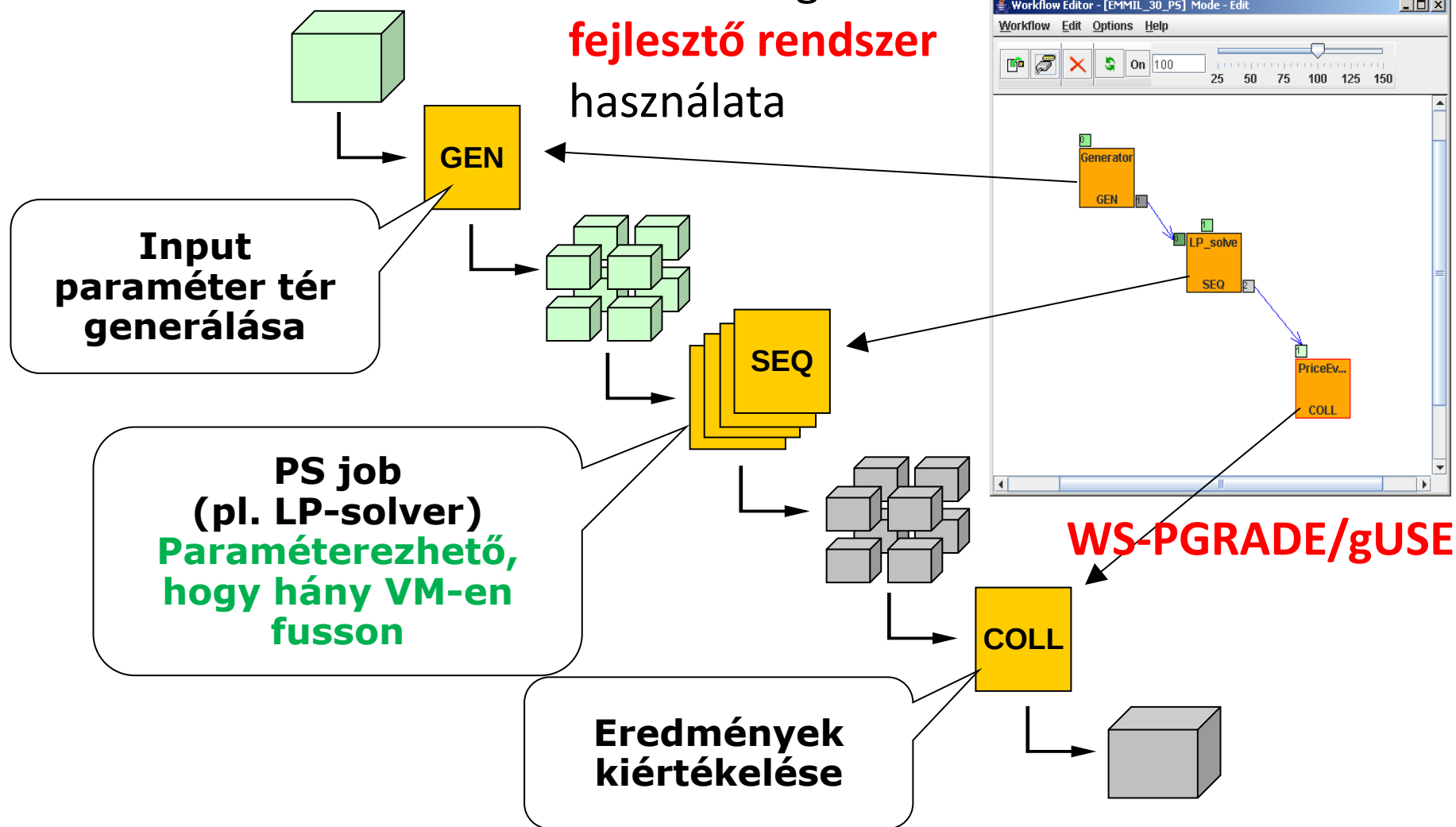
- Ezek közül a tiny-t nem támogatjuk, mert az túl kicsi.

Paraméter sweep (PS) alkalmazások végrehajtása HTC párhuzamosítással



Paraméter sweep (PS) alkalmazások végrehajtása HTC párhuzamosítással

2. lehetőség: **PaaS fejlesztő rendszer** használata



2.1 MPI alkalmazás végrehajtása



- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő **MPI futtató rendszert** tartalmazó Linux vagy Windows gép gyors létrehozása a felhőben
 - lemezképet adunk, amiből a kívánt gép indítható
 - Négyféle mérettel lehet ezeket elindítani (small és medium ajánlott teszteléshez, large és xlarge ajánlott futtatáshoz)
 2. MPI alkalmazás telepítése a létrehozott gépen vagy beküldése a gépre
 3. A létrehozott gépről elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen
- Ez már egy komplexebb IaaS alkalmazás, aminek a lemezképét később fogjuk közzé tenni az MTA Cloud web lapon

2.2 OpenMP alkalmazás végrehajtása



- Igények és kielégítésük az MTA Felhőben:
 1. Nagyon hasonló az MPI-hoz
- Ez is egy komplexebb IaaS alkalmazás, aminek a támogatását később fogjuk közzé tenni az MTA Cloud web lapon

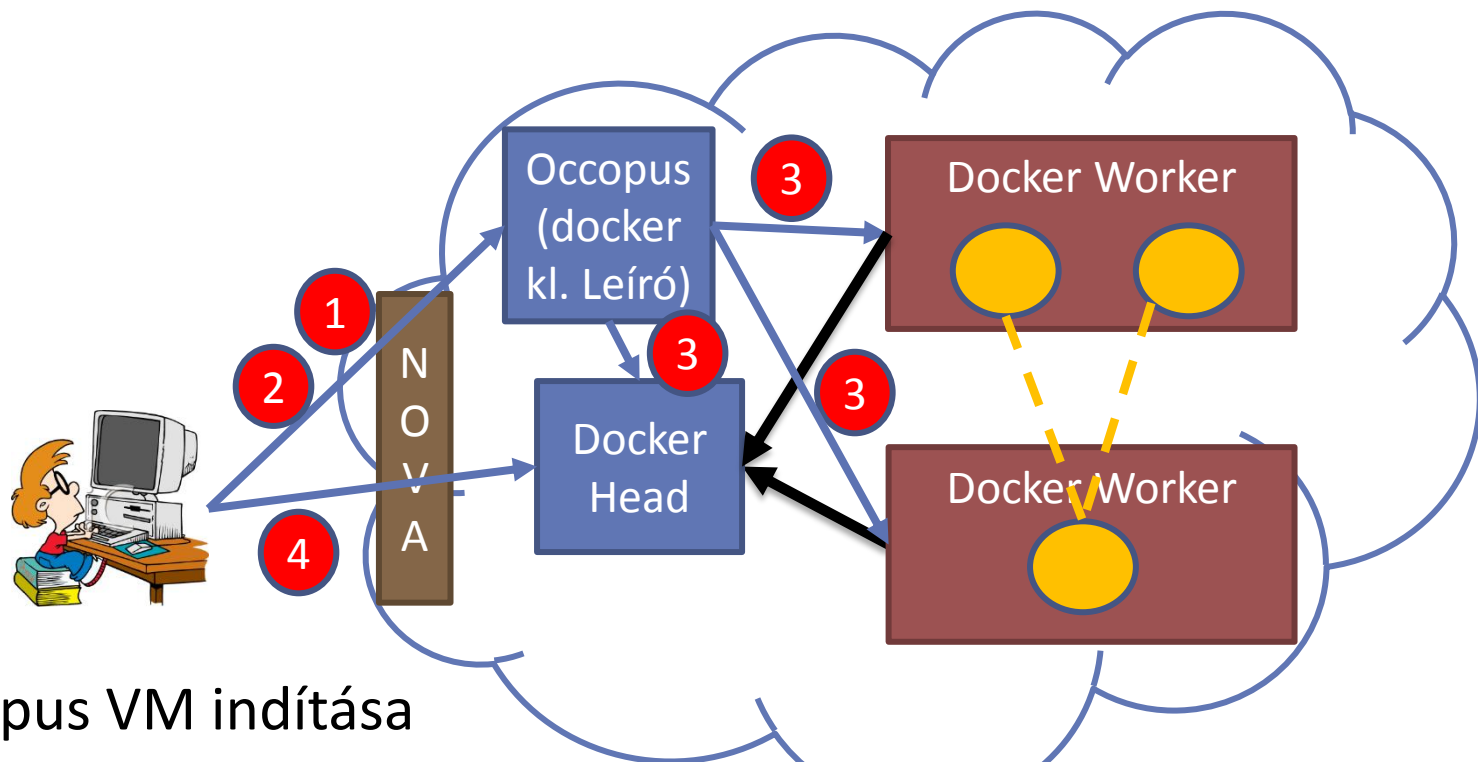
3.1 Docker alkalmazás végrehajtása párhuzamosítás nélkül



- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő, **docker konténert tartalmazó** Linux vagy Windows gép gyors létrehozása a felhőben
 - Linux és Windows lemezképeket adunk, amikből a kívánt gépek indíthatóak
 - Négyféle mérettel lehet ezeket elindítani
 2. szekvenciális alkalmazás telepítése a docker konténerben
 3. A docker konténerből elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott gépen

docker klaszteren

- Igények és kielégítésük az MTA Felhőben:
 1. Az alkalmazás méretétől függő, **docker klaszter** létrehozása a felhőben
 - A szükséges lemezképeket, Occopus szervízt és Occopus fájl leírókat adjuk. Ld.:
<http://occopus.lpds.sztaki.hu/tutorials;jsessionid=0F6D68DAA04639B99D039D6FBE9F2A6B>
 2. Belépés a Docker klaszter master node-ra és a Docker alkalmazás telepítése docker konténerben
 3. A docker konténerből elérni a tárolót, ami a feldolgozandó adatot tárolja
 4. Futtatni az alkalmazást a létrehozott docker klaszteren



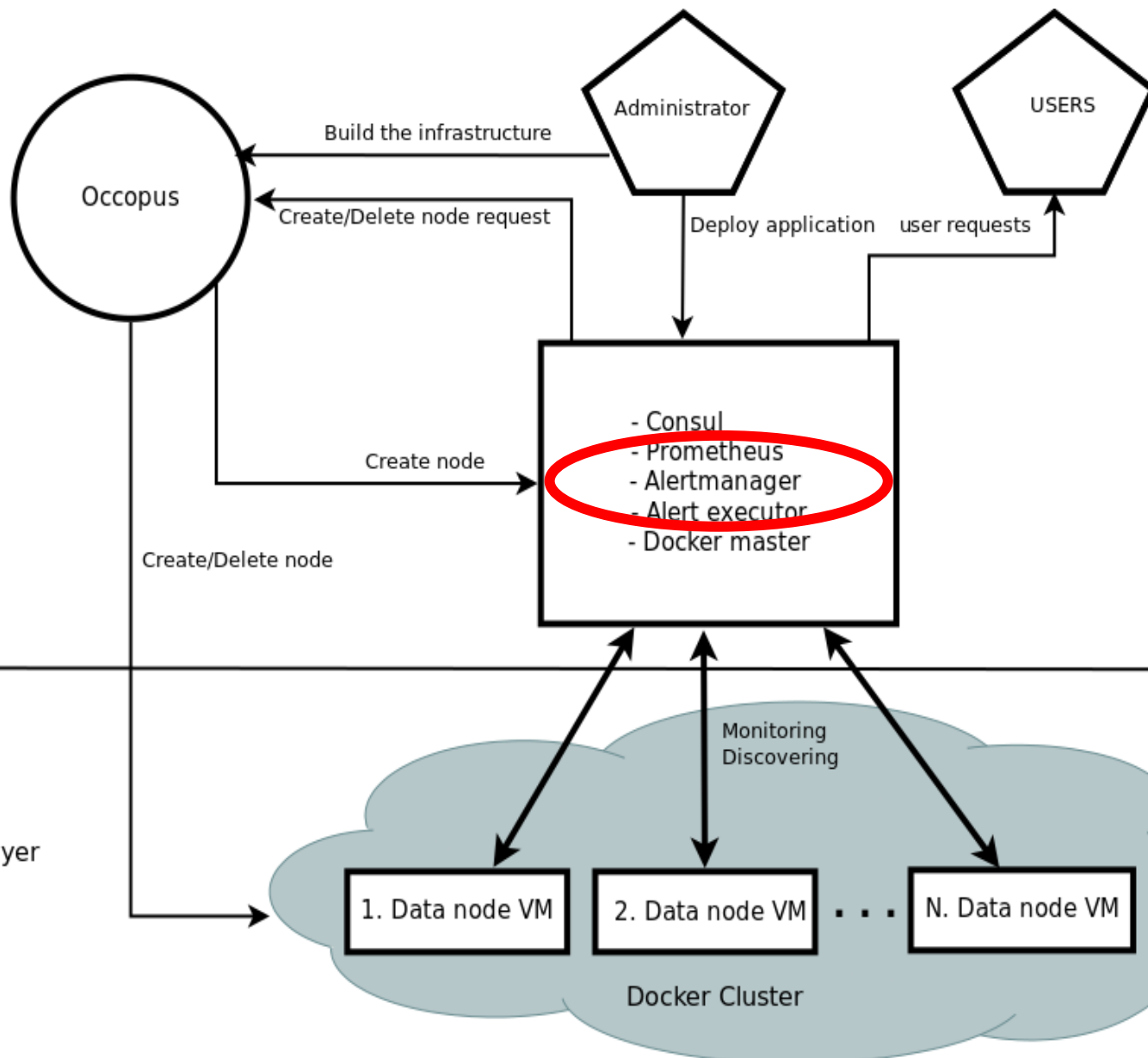
1. lépés: Occopus VM indítása

2. lépés: Belépés az Occopus VM-re és a Docker Klaszter Leíróval az Occopus indítása

3. lépés: Occopus automatikusan létrehozza a Docker Klasztert a felhőben

4. lépés: Belépés a Docker Klaszterbe és használata

Skálázható docker klaszter felépítése



- Ld. Nagy Enikő előadásában

5.1 Egyedi pipeline alkalmazás végrehajtása

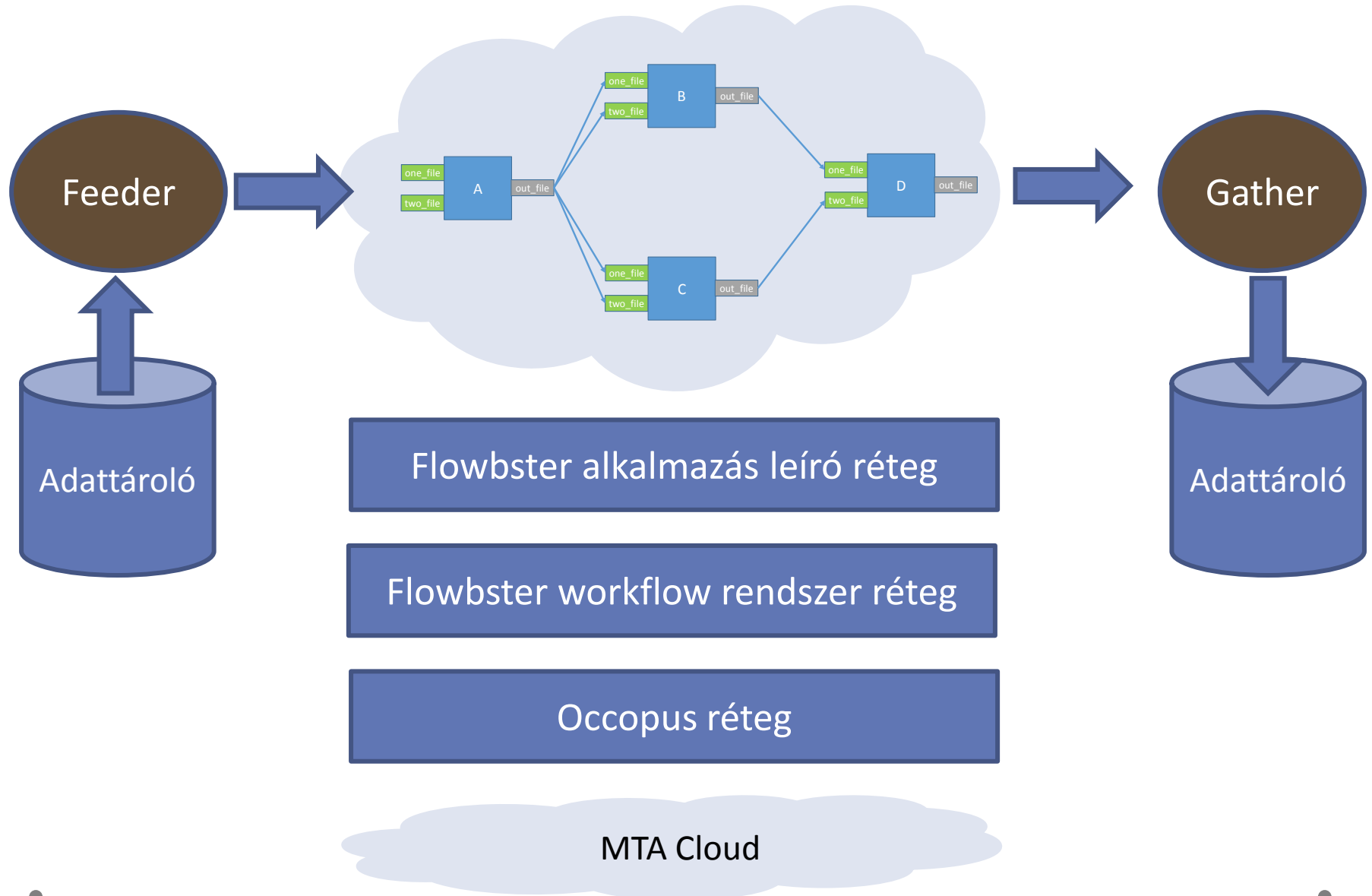
- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (pl. WS-PGRADE/gUSE)
 1. Az alkalmazás pipeline definiálása, mint workflow
 2. Az alkalmazást a bemenő adatokkal beküldeni a felhőbe

5.1 Egyedi
pipeline:

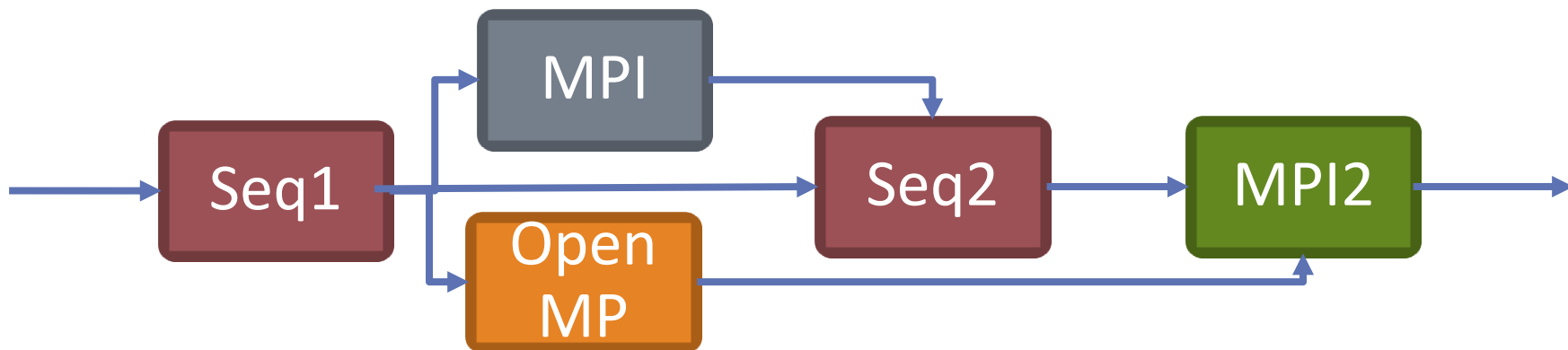


- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (Flowbster, ami Occopus-ra épül)
 1. Flowbster (pipeline) node-ok és kapcsolatuk definiálása a Flowbster alkalmazás leíró réteg segítségével. Két opció:
 1. Szövegesen Occopusnak megfelelő fájl leíró nyelven (Occopus tudást igényel)
 2. Grafikusan (semmilyen előismeret nem kell hozzá) – fejlesztés alatt
 2. szekvenciális alkalmazások definiálása a pipeline node-ok számára (Flowbster node-ok konfigurálása)
 3. Szükséges virtuális gépek indítása a felhőben és összekapcsolásuk a Flowbster definíció szerint – Occopus végzi el
 4. Átfolytatni az adatokat a létrehozott pipeline-ba kapcsolt gépeken

Flowbster koncepció



6. Workflow alkalmazás végrehajtása

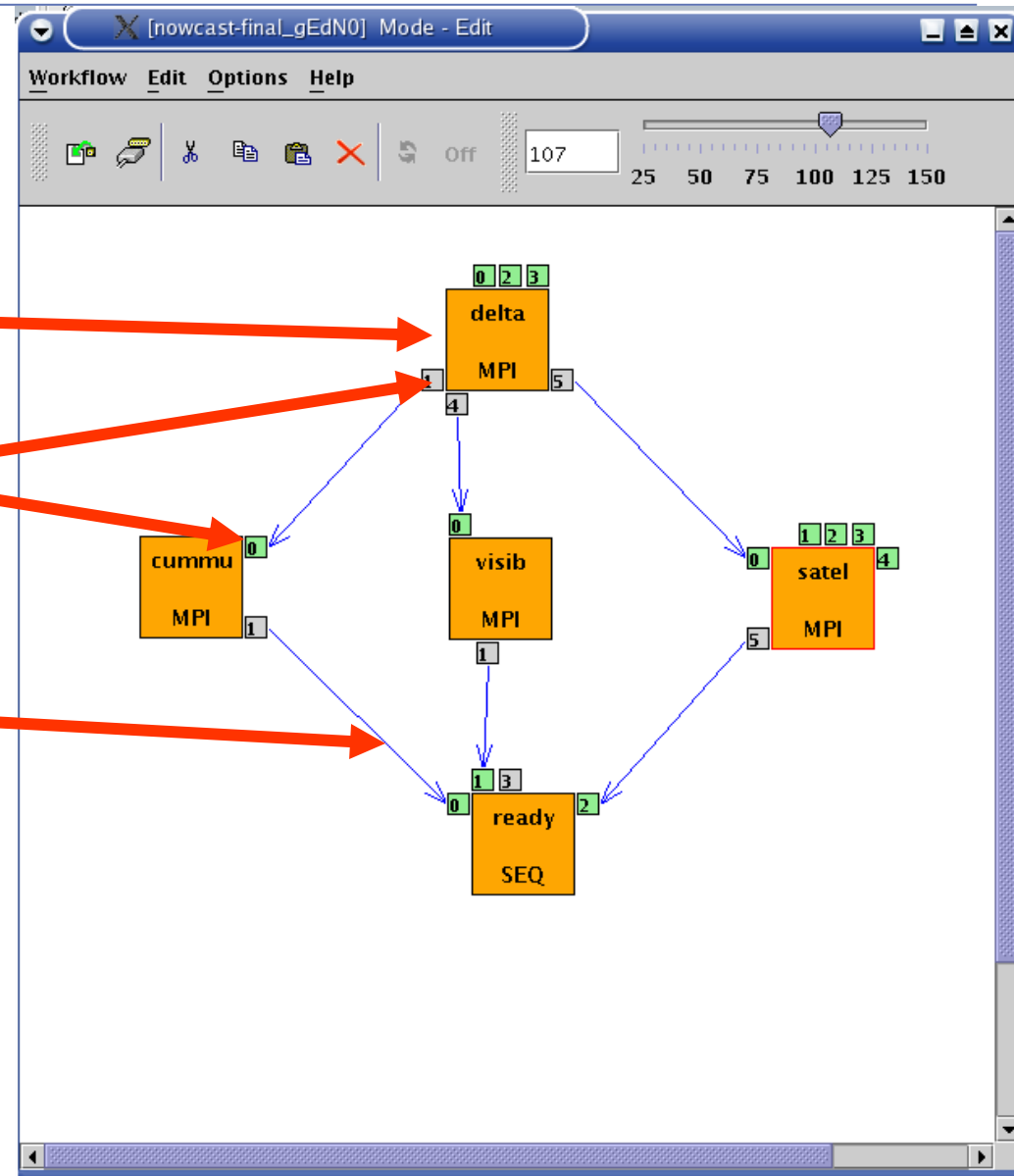


6. Workflow alkalmazás végrehajtása



- Két opció:
 - IaaS API segítségével a felhasználó programozza be
 - PaaS eszköz segítségével (WS-PGRADE/gUSE)
 1. Az alkalmazás definiálása, mint WS-PGRADE workflow
 2. Az alkalmazást a bemenő adatokkal beküldeni a felhőbe

- **Irányított aciklikus gráf,** ahol
 - A node-ok **job-okat, szervízeket, vagy beágyazott workflow-t** reprezentálnak
 - A port-ok input/output fájlakat reprezentálnak
 - Az élek a node-ok közötti adatmozgatást és függőséget reprezentálják
- **Workflow szemantika:**
 - Egy node tüzelhet, azaz a hozzá tartozó job végrehajtható, ha minden input fájl megérkezett



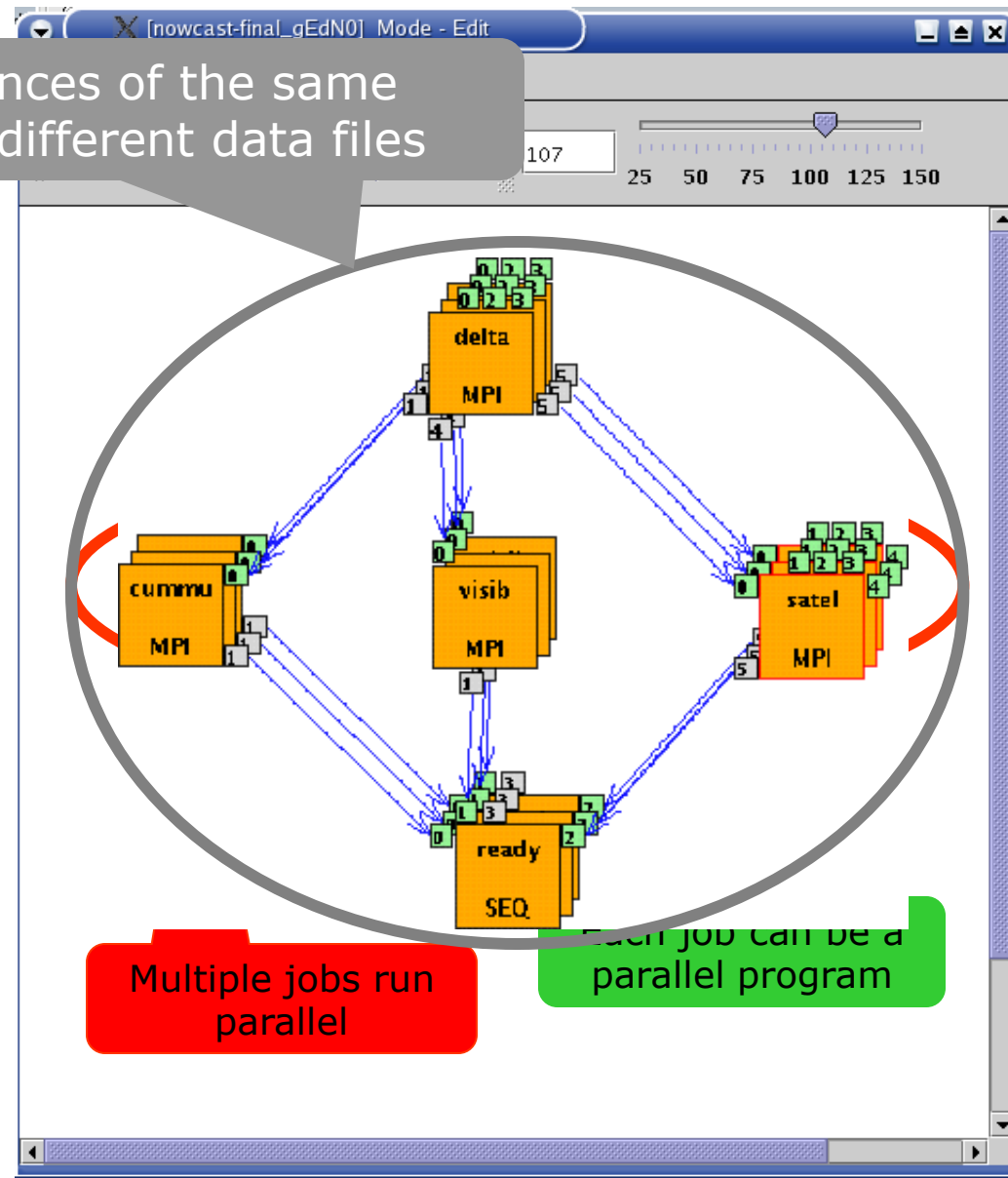
Három szintű párhuzamosság

Multiple instances of the same workflow with different data files

– Parallel execution inside a workflow node

– Parallel execution among workflow nodes

– Parameter study execution of the workflow



- A **standard OpenStack IaaS támogatás** készen áll, október 1-től elérhető a kész szolgáltatás
- Bonyolultabb IaaS VM szerkezetek (pl. docker klaszter, pipeline, stb.) létrehozásához érdemes az **Occopus** szolgáltatást használni
- Magasabb szintű alkalmazásokhoz érdemes **PaaS eszközöket** alkalmazni:
 - **WS-PGRADE/gUSE**: sokféle felhővel megy már, így az MTA Clouddal is
 - **Flowbster**: a prototípus rendelkezésre áll
- A Flowbster kivételével minden felsorolt és bemutatott eszköz elérhető és használható az MTA Cloudon