

Web2Grid: Desktop Grid a Web 2.0 szolgálatában

Marosi Attila Csaba
MTA SZTAKI
Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutató Intézet
H-1528 Budapest, Pf. 63.
Email: atisu@sztaki.hu

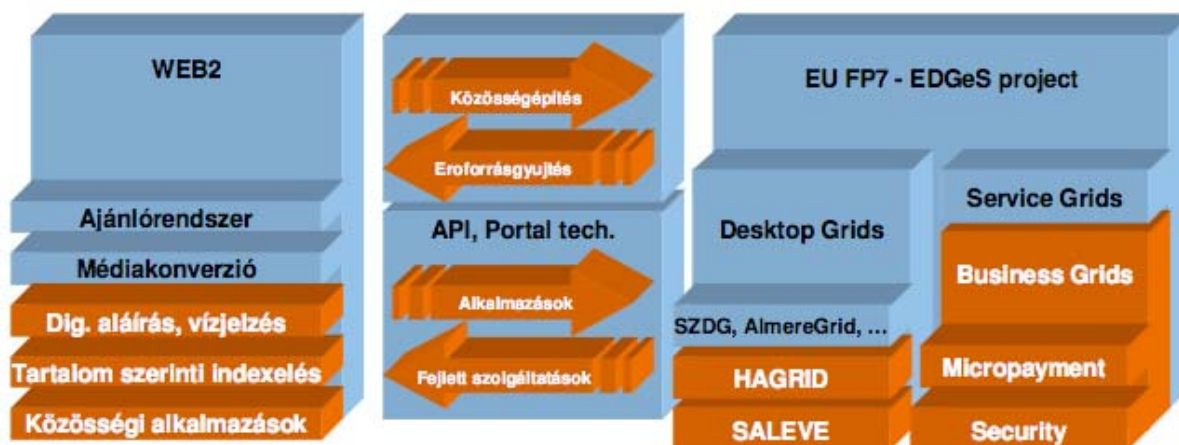
1. Bevezetés

A Grid rendszerek az elmúlt évtizedben alapvetően az akadémiai és egyetemi kutatásokhoz biztosítottak nagy léptékű elosztott számítási és tárolási kapacitást; így a korábban virtuális szervezetekbe tömörült felhasználók hatékony módon képessé váltak megosztani kutatási eszközeiket, informatikai tárolási és számítási erőforrásaikat. Az utóbbi években egyre inkább a szolgáltatások és a tudás megosztása került előtérbe, továbbá kritikus ponttá vált az alkalmazási területek kibővítése a rendszerek egységes elérhetőségének biztosításával, azaz a heterogenitás leküzdése is. A Grid oldaláról alapvetően ez által nemcsak a tradicionális szolgáltatás Gridok (mint például a legnagyobb európai Grid, a CERN által koordinált EGEE [12]), de a Web 2.0 filozófiájához közelebb álló Desktop Gridok (mint például a SZTAKI Desktop Grid) is egységes módon elérhetővé válnak az alkalmazásfejlesztők és végfelhasználók számára. A Desktop Gridok esetén gyakran közösségi felajánlás segítségével épül fel a Grid rendszer (pl. SETI@home), ezért is hasonlítható a Web 2.0 tartalomépítéséhez. A Web 2.0 kifejezés olyan második generációs internetes szolgáltatások gyűjtőneve, amelyek elsősorban a közösségre épülnek, azaz a felhasználók közösen készítik a tartalmat vagy megosztják egymás információit. A Web 2.0 szolgáltatásoknál a szerver gazdája csak a keretrendszert biztosítja, a tartalmat maguk a felhasználók töltik fel, hozzák létre, osztják meg vagy véleményezik. A felhasználók jellemzően kommunikálnak egymással, és kapcsolatokat alakítanak ki egymás között. Általában a nagyszámú kapcsolat és kommunikáló közösségi tag miatt igen nagymennyiségű adat és összetett relációkat kell kezelni a Web 2.0-as rendszerekben, ami rendkívüli feldolgozási kapacitást igényel. A Web 2.0 alkalmazások egy nagy része pedig már eleve nagyméretű fájlokat (pl.: videó megosztó) kezel; ezek konvertálása, különböző feldolgozása igen erőforrás igényes. Tehát a Web 2.0 oldaláról több nagy informatikai kapacitást igénylő feladat megoldása során is felmerül Grid alkalmazásának lehetősége; például a már manapság is elérhető ajánlórendszerek, vagy médiakonvertálási feladatok kapcsán, de a várható igények között szerepel pl. a digitális aláírás és vízjelezés, de számos közösség-specifikus alkalmazás is prognosztizálható a közeljövőben. Az eddigi folyamatok azt mutatják, hogy a felhasználói igények hatására folyamatosan növekvő teljesítmények, ill. információs bázisok újabb és újabb felhasználási ötleteket, lehetőségeket generálnak miközben az elérhető szolgáltatások komplexitása egyre nő ezzel egy gyorsuló spirált alkotva.

A cikk felépítése a következő. A második fejezetben a Grid és Web 2.0 technológiák integrálását célul kitűző Web2Grid [1] [2] projektet mutatom be, majd a projekt számos eredménye közül a BOINC desktop grid rendszer különböző kiterjesztéseit részletezem. Végül összefoglalom a projekt eddigi eredményeit és ismertetem a további célkitűzéseit.

2. A Web2Grid projekt

A projekt elsődleges célja az MTA SZTAKI által koordinált EU FP7 EDGeS [3] [4] projekt és az ECONET által koordinált HAGRID [5] projekt desktop gridhez kötődő eredményeinek felhasználásának elősegítése különös tekintettel a Web 2.0 technológiák bevonásával. A Web 2.0-es megoldások egyrészt elősegítik az erőforrások önkéntes vagy elszámolás alapon történő biztosítását a desktop gridbe („közösség ereje”). Másrészt a több hazai fejlesztésű desktop grid rendszerrel (BME, ECONET, SZTAKI [9] [10]) is kibővített platform informatikai háttér-infrastruktúrát biztosítják a Web 2.0 portálok üzemeltetésekor fellépő szokásos igényekhez (nagy számú felhasználó/kapcsolat/fájl kezelése, nagyméretű fájlok konvertálása), az új biztonsági követelmények kielégítéséhez (vízjelzés, digitális aláírások időszakos felülvizsgálata, stb.), valamint a nagy informatikai kapacitást igénylő és a közösségi tagok által definiált feladatok végrehajtásához. A Web 2.0 és a Desktop Grid technológiák ötvözése mindkét terület számára számos előnyt biztosít. A Web 2.0 így már nemcsak a közösségi tartalmak, hanem a szolgáltatások megosztása felé mozdulhat el a Grid technológiák segítségével. A projektben a fenti feladatok vizsgálata, illetve a feladatokat ellátni képes szolgáltatások, eszközök és interfészek kifejlesztése, valamint a szükséges módszertani lépések kidolgozása a cél mind zárt (intézeti desktop grid), mind nyitott környezetben (nyilvános desktop grid). A projekt különösen nagy hangsúlyt fektet a szolgáltatás és üzleti Grid felhasználásának elősegítésére a biztonság és elszámolás kérdéseire fókuszálva, az E-Group megoldásaira alapozva. A kidolgozott megoldások demonstrálása céljából a konzorcium több területről mutat fel Web 2.0 alapú mintaalkalmazásokat, valamint könnyen használható interfészeket és ajánlásokat készít minél szélesebb körű felhasználás előmozdítására.

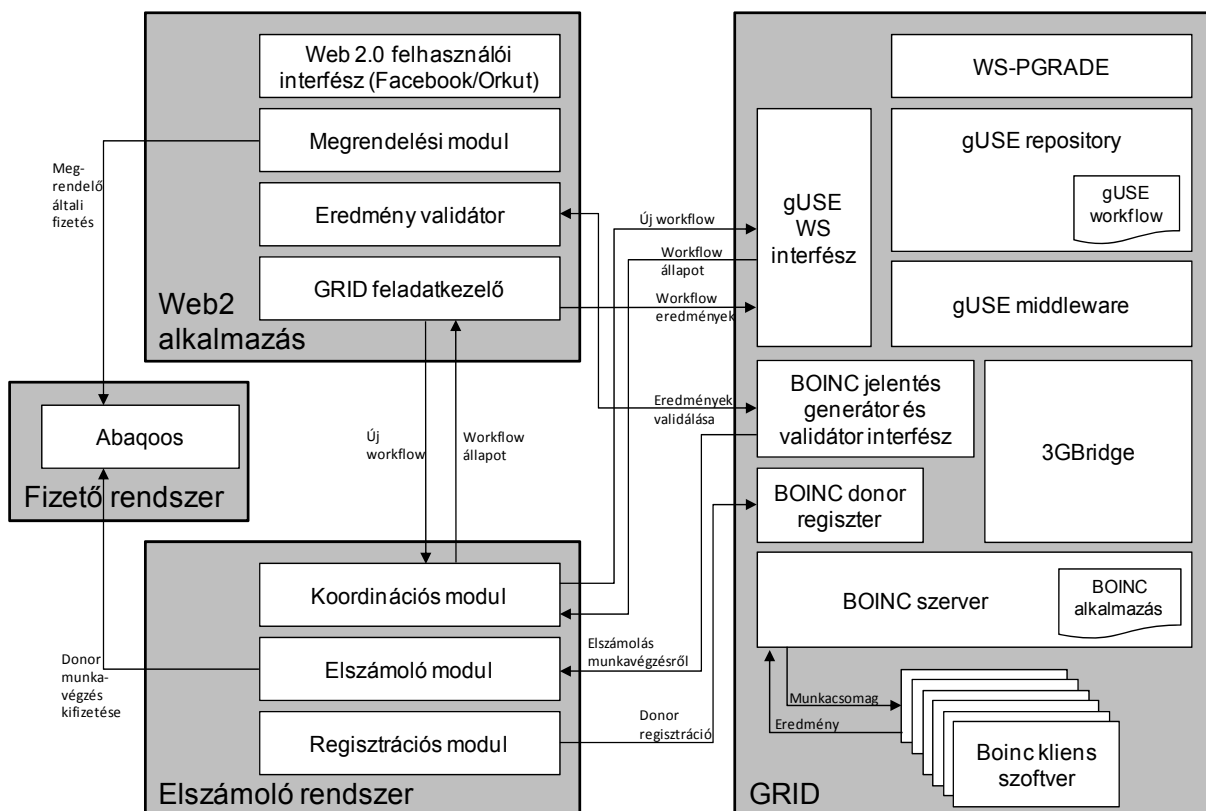


Ábra 1 A Web2Grid projekt célkitűzései

A Web 2.0 és Grid közötti kapcsolódást alapvetően jól definiált alacsony szintű alkalmazásfejlesztési interfészen keresztül (API) és magas szintű alkalmazásfejlesztő grafikus eszközök kidolgozásával biztosítják a partnerek. Az interfészek kidolgozása során figyelembe kell venni, hogy a Web 2.0 alkalmazások különböző területekről jöhetnek; maguk a Web 2.0 szolgáltatók, külső szakértők (pl. biztonsági területek esetén), de a közösség tagjai is felléphetnek alkalmazásokkal (ld. például a Facebook már elérhető API-jait). Ez indokolja a többszintű interfész kidolgozását, ahol projekt számos EU 6. és 7. keretprogramban futó projektben kidolgozott technológiára alapoz. Az interfészek kidolgozásakor különös hangsúlyt fektettek a partnerek a protokollokra, amely a biztonsági és elszámolási funkciók megvalósítását teszi lehetővé. A projekt során kiemelt jelentőséggel bír, hogy üzleti grid irányra is fókuszáljon, két területen: biztonság és elszámolás (micropayment), melyek szintén elősegítik a két terület (Web 2.0 és Grid) előnyeinek ötvözését.

3. A BOINC rendszer fejlesztései a projekt keretében

A Web2Grid egyik célkitűzése, hogy a Web 2.0-es számításigényes feladatokat közösségi erőforrás-felajánlás segítségével hajtassa végre, ehhez pedig a nyitott (globális) desktop grid-ek erejét használja fel. Az egyik legelterjedtebb desktop grid rendszer a BOINC [6] [7], ezért projekt által a kialakított rendszer - a hazai fejlesztésű rendszerek mellett - ezt támogatja. A BOINC-ot támogató kialakított architektúrát az Ábra 2 mutatja be.



Ábra 2 A Web2Grid BOINC-ot és a hozzá kapcsolódó modulokat tartalmazó keretrendszer-részének felépítése

A „WEB2 alkalmazás” (az Ábra 2 –n), melyből tetszőleges mennyiségű lehet a rendszerben, többek között tartalmaz egy az adott közösségi oldalhoz (Facebook, Orkut) tartozó felületet a felhasználóknak, valamint egy eredmény validátor-t, mely a Grid rendszer által elvégzett feladatok helyességét ellenőrzi. Minden ilyen validátor, alkalmazás specifikus, mivel csak az alkalmazás készítője tudja meghatározni azokat a feltételeket, ami alapján a végrehajtott feladatról eldönthető, hogy sikeres volt-e vagy nem.

A végrehajtott Grid rendszerben egy munkafolyamat ütemező rendszer (gUSE) vezényli az alkalmazás futását. Ez a 3G Bridge nevű komponens segítségével adja át az aktuális feladatokat a desktop grid számára. A 3G Bridge lehetővé teszi feladat (taszk) szinten különböző Grid-ek együttműködését, így lehet a többi (desktop) gridet a rendszerhez illeszteni. Jelen esetben a feladatokat a BOINC rendszer hajtja végre, önkéntesen felajánlott erőforrások (kliensek) segítségével. Ehhez azonban szükséges a futtatandó alkalmazásnak, egy speciálisan a BOINC platformra portolt változata.

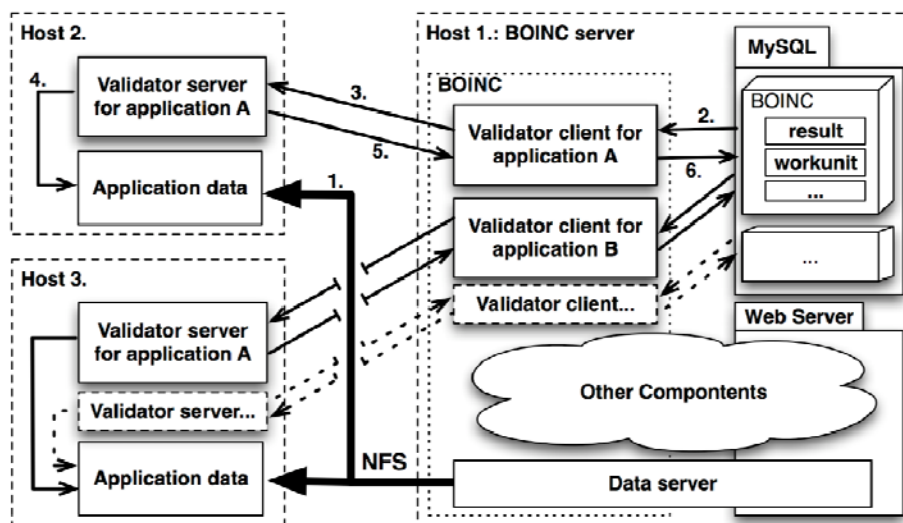
E kialakított rendszernek láthatóan több problémát és feladatot kell megoldania, a továbbiakban ezeket részletezem.

3.1 Feladatok távoli ellenőrzése

Az egyik probléma az önkéntes erőforrás-felajánlásokon alapuló nyitott desktop grid rendszerek egyik sajátossága, hogy a feldolgozott, avagy kiszámolt feladatok helyességét ellenőrizni kell. Az ezt végző validátor komponens a desktop grid szerveren fut, mivel a desktop grid szerverre küldik vissza a kliensek az eredményeiket, bár valójában a (WEB2) alkalmazás része. Ez azt jelenti, hogy egy harmadik fél által telepített (az alkalmazáskészítője által írt, azaz nem megbízhatónak minősülő) komponens fut a szerveren, ami ráadásul logikailag sem a szerverhez, hanem az alkalmazáshoz tartozik. A kiszámolt feladatok ellenőrzése kétféleképpen történhet: a kimeneti fájlokon formai ellenőrzés, azaz a formátuma megfelel-e az elvárásoknak, vagy a másik lehetőség, hogy ugyanazon feladatot több példányban küldik ki különböző klienseknek (redundancia), és a visszakapott eredményeket hasonlítják össze. Mind a formai ellenőrzés, mind az eredmények összehasonlítása alkalmazás specifikus feladat (pl.: az alkalmazástól függ, hogy mikor tekinthető két eredmény egyezőnek).

A Web2Grid projekt célkitűzése, hogy minél több Web 2.0 alkalmazás számára tegye elérhetővé a Grid technológiákat, azonban ez nem mehet a biztonság rovására, így szükségessé vált minden harmadik fél által írt, és a szerveren futtatott komponens eltávolítása, a desktop grid szervertől való szeparálása olyan módon, mely garantálja a biztonságot, de ugyanakkor transzparens megoldást nyújt az alkalmazás- és validátor-fejlesztők számára is.

Az Ábra 3 mutatja a módosított eredmény-ellenőrző (validátor) rendszer működését. Az eddigi monolitikus felépítést [8] felváltotta egy szerver-kliens architektúra. A BOINC szerver csak validátor klienseket tartalmaz, melyek távoli validátor szerverekhez csatlakoznak a rendszer által elvégzett feladatok ellenőrzése miatt. A validátor kliens általános, míg a szerver tartalmazza az alkalmazás specifikus részt. Az alkalmazás készítőknak így egy szerver komponensre kell létrehozniuk, melyhez adott egy keretrendszer, így csak az alkalmazásukhoz tartozó specifikus függvényeket kell implementálni. Az elkészült szerver komponens később tetszőleges gépen futhat, a desktop grid rendszer veszi fel vele a kapcsolatot, amint eredményt kell ellenőrizni.



Ábra 3 A BOINC módosított validátor rendszere

Az ellenőrzés folyamata a következőképpen zajlik. Első lépésként (Ábra 3, 1. lépés) a távoli validátor szervernek read-only hozzáférést kell kapnia a BOINC adatszerveréhez, mivel az eredmények ellenőrzéséhez tartozik a kimeneti fájloknak az ellenőrzése. Ez többféleképpen lehetséges, például hálózati megosztással (NFS – Network File System) vagy pedig a fájlokhoz közvetlen hálózati hozzáféréssel (például Parrot [11] segítségével). Adatbiztonság szempontjából fontos, hogy a validátor szerver csak a saját ellenőrizendő taszkjaihoz tartozó ki és bemeneti fájlokat érheti el, és azokat is csak olvasásra.

A validátor kliensek folyamatosan figyelik a BOINC adatbázisát, hogy mikor kerül egy taszk feldolgozott és ellenőrzésre váró állapotba (Ábra 3, 2. lépés). Ilyen esetben továbbítja a megfelelő validátor szerver felé a taszk adatait (3. lépés). Fontos megjegyezni, hogy nagyszámú feladat továbbítása szükséges lehet, ezért mindenképp állandó kapcsolat és kötegetelt adatküldés szükséges a szerver és kliens között, mivel a kapcsolat folyamatos felépítése és bontása idő és processzor-igényes feladat lenne. A validátor szerver a taszk(ok) megkapott adatai és a kimeneti fájllai alapján eldönti, hogy melyek érvényesek és melyek hibásak (4. lépés). A validátor szerver csak memóriában dolgozik, és minden eredmény továbbít a kliens felé (5. lépés), majd pedig a kliens az eredményt az adatbázisba rögzíti (6. lépés).

3.2 Elszámolás

A projekt úgy próbál minél több erőforrást gyűjteni, hogy az elvégzett feladatokért virtuális kreditet oszt a felhasználóknak, ez azonban szükségessé teszi egy elszámoló modul kialakítását, mely a felhasználók felé jelenti az általuk megszerzett kreditek számát. A BOINC tartalmaz egy ilyen elszámoló rendszert, azonban ez azon alapszik, hogy a kliensek jelentik le az általuk elvégzett munka mennyiségét, ezért ez nem használható. A kialakított elszámoló rendszer felépítése szintén az Ábra 2 –n látható. Ez három modult foglal magában. A koordinációs modul feladata hogy a Grid rendszerben futó taszkokat egy-egy a

Web 2.0 felhasználói felületen indított feladathoz rendelje. Például egy felhasználó szeretne több képet vízjelezni, ilyenkor az egyes képek külön taszkként kerülnek a Grid rendszerbe, de ezeket nyomon kell követni, hogy a felhasználóhoz a taszkok egyértelműen hozzárendelhetőek legyenek a későbbi elszámolás végett. Az elszámoló modul feladata a végrehajtott taszkok alapján egyrészt a felhasználók felé jelentés (az esetleges számlázás miatt), illetve a kliensek által végzett munka nyilvántartása és ez alapján virtuális pénz („kredit”) jóváírása. Végül a regisztrációs modul feladata, hogy a Web 2.0 felhasználói interfészen regisztrált erőforrás donorokat a desktop grid rendszerbe is regisztrálja, illetve a két rendszer felhasználó bázisát szinkronban tartsa. Ezt a következő alfejezetben részletezem.

3.3 Regisztráció: erőforrás donorok és felhasználók

A rendszer a Web 2.0 közösségi platformokon (például Facebook) szerez erőforrás-donorokat és felhasználókat is. Erőforrás-donorok azok, akik a gépük szabad kapacitását felajánlják taszkok (például vízjelezés) futtatására, felhasználók pedig az ezeket a feladatokat „megrendelők”: például valamelyik Facebook-ra feltöltött albumuk képeit szeretnék vízjeleztetni.

A regisztráció, donoroknak és felhasználóknak is, a Web 2.0 közösségi oldalon történik. Felhasználóként az adott Web2Grid alkalmazás Web 2.0 felhasználói interfészén kell regisztrálni, donorként pedig közvetlenül a Web2Grid erre kialakított felületén. Mivel a Web2Grid több desktop grid rendszert tartalmaz, illetve az elvégzett munkát saját maga naplózza (elszámoló modul) ezért nem lehetséges közvetlenül a desktop gridekben történő regisztráció, a donorok felé a használt rendszer felülete teljes mértékben rejtve marad. Ezért a regisztrált donorokat a központi elszámoló részeként felállított modul szinkronizálja saját maga és a különböző desktop gridek között.

4. Összefoglalás

A Web2Grid platform jelenleg zártkörű tesztelése folyik, illetve több alkalmazás előkészítése és tesztelése zajlik. A konzorcium célja, hogy minél több alkalmazást és szolgáltatást tegyen elérhetővé a Web 2.0 közösségi platformok segítségével.

A bemutatott eredmények a Nemzeti Kutatási és Technológiai Hivatal (NKTH) támogatásával, a TECH_08-A2/2-2008-0097 számú támogatási szerződés (WEB2GRID projekt) keretein belül, illetve az Európai Unió 7. Keretprogramjának (FP7/2007-2013), 261556-es szerződésszám alapján kapott támogatása segítségével jöttek létre.

Hivatkozások

[1] Web2Grid. <http://web2grid.econet.hu/>

[2] Lovas et al. 2009. Supporting new generation community WEB services and applications by GRID platform. Networkshop 2009.

[3] Enabling Desktop Grids for e-Science (EDGEs). <http://www.edges-grid.eu/>

[4] Urbah, E., Kacsuk, P., Farkas, Z., Fedak, G., Kecskeméti, G., Lodygensky, O., Marosi, A. Cs., Balaton, Z., Caillat, G., Gombás, G., Kornafeld, A., Kovács, J., He. H., Lovas., R.: EDGeS: Bridging EGEE to BOINC and XtremWeb. *Journal of Grid Computing*, 7/3. 335-354. 2009.

[5] HAGRID. <http://hagrid.econet.hu/>

[6] Berkeley Open Infrastructure for Network Computing (BOINC). <http://boinc.berkeley.edu/>

[7] D. P. Anderson, "BOINC: A system for public-resource computing and storage," in *Fifth IEEE/ACM International Workshop on Grid Computing*, R. Buyya, Ed., 2004, pp. 4–10.

[8] BOINC validátor keretrendszer. <http://boinc.berkeley.edu/trac/wiki/ValidationSimple>

[9] SZTAKI Desktop Grid. <http://www.desktopgrid.hu/>

[10] A. Csaba Marosi, G. Gombás, Z. Balaton, P. Kacsuk, and T. Kiss, "Sztaki Desktop Grid: Building a scalable, secure platform for desktop grid computing." in *Making Grids Work*. Springer Publishing Company, Incorporated, July 2008, pp. 363–374.

[11] Douglas Thain and Miron Livny, Parrot: Transparent User-Level Middleware for Data-Intensive Computing, *Workshop on Adaptive Grid Middleware*, New Orleans, Louisiana, September 2003.

[12] Enabling Grids for E-science (EGEE). <http://eu-egee.org/>