

# Peer2Meet: Csoportmunka támogatás és workflow kezelés Peer2Peer rendszerekben

Fülöp Csaba, Kovács László  
[csabi@dsd.sztaki.hu](mailto:csabi@dsd.sztaki.hu), [laszlo.kovacs@sztaki.hu](mailto:laszlo.kovacs@sztaki.hu)  
MTA SZTAKI  
Elosztott Rendszerek Osztály

## ***Kivonat***

A Peer2Meet projekt célja egy csoportmunka rendszerek alapjául felhasználható alrendszer valamint egy erre épülő elektronikus önkormányzati (közgyűlési) rendszer elkészítése volt. A rendszer egy peer2peer csomagra épül, amely egy elosztott objektum tárat valósít meg. Az egyes csomópontoknál a tárba tett objektumokat a rendszer a peer2peer hálózaton eljuttatja a többi csomóponthoz. A megfelelő biztonság érdekében minden objektum titkosított, melyhez csak a megfelelő személyek férnek hozzá. Egy elektronikus közgyűlési rendszerhez szükséges objektum típusok például: dokumentum, határozat, szavazás, üzenőfal, stb. Az elosztott rendszeren egy workflow rendszert is létrehoztunk. Az egyes csomópontokon egy workflow ütemező fut, és ezen ütemezőket a csomópontok üzenetek segítségével tartják szinkronban.

## ***Bevezetés***

A Peer2Meet projekt célja egy csoportmunka rendszerek alapjául szolgáló elosztott architektúra elkészítése volt. Középpontban a laikus felhasználók számára nyújtott egyszerű használhatóság állt. Az architektúra lehetőségeinek bemutatására elkészítettünk egy elektronikus közgyűlési rendszert is, önkormányzati felhasználás céljára.

A csoportmunka és dokumentumkezelés támogatására jelenleg is számos lehetőség áll a felhasználók rendelkezésére. A megfelelő rendszer kiválasztása természetesen a felhasználási igényektől függ. A Peer2Meet rendszer elkészítésével egy újfajta ötlet kidolgozása, működőképességének bemutatása volt a célunk.

## ***A rendszer felépítése***

A rendszer peer2peer hálózaton kommunikáló csomópontokból áll, így nem szükséges központi szerver az üzemeltetéséhez. Az objektum tár teljes másolata megtalálható minden csomóponton. Ez a napjainkban rohamosan növekvő merevlemez kapacitásoknak köszönhetően nem jelent szigorú követelményt a felhasználók számára. Az objektumok megfelelő titkosításáról természetesen gondoskodik a rendszer, így az objektumok csak kódolt formában találhatók meg a gépeken és a hozzáférés csak a szükséges jogosultsággal rendelkező felhasználók számára lehetséges. Az objektum táruk szinkronizálása automatikusan, a felhasználók számára láthatatlanul történik. Ez a folyamat gondoskodik a lokálisan keletkező új objektumok eljuttatásáról a többieknek és a távol keletkező új objektumok beszerzéséről. A bonyolultabb folyamatok segítésére és a

felhasználók tevékenységének vezetésére egy workflow alrendszert is tartalmaz a rendszer.

<b>Alkalmazás specifikus objektumok</b>	<b>Workflow rendszer</b>
<b>Alapvető objektumok:</b> dokumentum, chat, szavazás, ...	
<b>Objektum tár:</b> titkosítás, szinkronizálás	
<b>peer2peer hálózat</b>	

1. ábra, a rendszer felépítése

## **Az objektumok terjesztése**

Alapesetben, ha folyamatos hálózati kapcsolatban állnak a csomópontok, akkor adott időközönként szétküldik egymás számára az objektum listájukat. A többi csomópont ezt megkapva megvizsgálja, hogy a saját objektum tárához képest van-e új objektum a listában. Ha van, akkor elkéri azt a lista küldőjétől. Így az új objektumok terjesztése legkésőbb a következő lista szétküldéskor megkezdődik a belépett felhasználók számára.

Hálózati kapcsolat kiesése esetén az adott csomópont szinkronizálása a többiekkel átmenetileg szünetel. A kapcsolat helyreállása után azonban ismét felveszi a kapcsolatot a többi csomóponttal és azok ismét elkezdik számára küldeni az objektum listáikat. A listákból a csomópont kiszűri a kiesése alatt keletkezett objektumokat és beszerzi azokat.

A hálózati kapcsolat kiesése azonban nem gátolja a tárral való munkavégzést, azaz a kiesett csomóponton dolgozó felhasználó továbbra is hozzáfér az objektumaihoz és hozhat létre újakat is. Így a kapcsolat helyreállása esetén a fenti folyamat játszódik le a fordított irányban is, azaz a hálózat többi tagja is értesül a kiesett csomópont listájából a kiesése alatt keletkezett objektumokról és beszerzik azokat.

A rendszer állapotát kizárólag a tár tartalma határozza meg és a csomópontok csak a tárba tett objektumokon keresztül kommunikálnak. Így, a fenti módszernek köszönhetően, egyetlen csomópont sem marad le egy üzenetről sem.

## **Backup, Archiválás**

Az objektum táruk fent leírt működése a backup és archiváló funkciók egyszerű megvalósítását teszi lehetővé.

Mivel minden objektum megtalálható minden csomóponton, így ez gyakorlatilag egy automatikus backup megoldásnak tekinthető. Ha valamely csomópont tönkremegy, akkor csak az utolsó szinkronizálása óta keletkezett objektumai vesznek el, az összes többi objektum megtalálható minden csomóponton, amelyeket helyreállítás után letölthet a többiektől.

Annak ellenére, hogy az objektum táruk méretét nem tekintettük szűk keresztmetszetnek, biztosítottunk lehetőséget, hogy a felhasználók korlátozzák a tár méretét (pl. mobil eszközök, stb.). Ekkor a tár az objektum listát ugyanúgy vezeti, de a megadott korlát átlépése esetén a legrégebbi objektumokat letörli és ezt az objektum listájában jelezve biztosítja, hogy a következő szinkronizáláskor ne próbálja meg beszerezni. Ha egy csomóponton ez a funkció ki van kapcsolva és a tárról rendszeres biztonsági mentés készül, akkor ezzel az archiváló funkció valósíthatjuk meg.

## **Az objektumok módosíthatósága és a funkciók modellezése**

Annak érdekében, hogy a hálózat magas fokú partíciónálódását elviselje a rendszer a tárba tett objektumok módosítása nem megengedett. Máskülönben a hálózat elveszítené rugalmasságát vagy pedig igen nehéz lenne egy elosztott döntési algoritmust értelmezni.



2. ábra, a chat objektumok hierarchiája

Természetesen egy olyan rendszer amiben nem lehet az adatokat módosítani elfogadhatatlan lenne a felhasználók számára. Ezért a rendszer egyes funkcióit több objektummal kell modellezni, oly módon, hogy az objektum hierarchiában a módosításokat alárendelt módosító objektumok ábrázolják. Például egy chat áll a chat szobából és üzenetekből (2. ábra). Így az új hozzászólásokért a chat szoba objektumot

nem kell módosítani, csak új üzenet objektumokat kell a hierarchiában létrehozni. Hasonló módosító objektumokkal van lehetőség egyes objektumok tartalmának módosítására esetleg törlésre is. A törlés ténye azonban csak a felhasználói felületen látszódik, a régi objektumok továbbra is a rendszerben maradnak és visszakereshetők.

## ***Biztonság***

A rendszer biztonsága és az érzékeny adatok védelme különösen fontos, mivel minden objektum megtalálható az összes csomóponton. Ezért, hogy csak jogosult felhasználók férjenek hozzá az adatokhoz, az objektumok titkosított formában találhatóak meg az objektum tárházban.

A titkosítás két lépésben történik. Az első lépésben az objektum titkosítása egy szimmetrikus kulccsal történik és így kerül az objektum tárházba. Majd nyilvános kulcsú titkosítás használva a jogosult felhasználók nyilvános kulcsával titkosítja a rendszer az előbbi szimmetrikus kulcsot. Majd az így kapott titkosított kulcs is az objektum tárházba kerül, mint az eredeti objektumhoz kapcsolódó módosító objektum. Ezt csak az adott nyilvános kulcs privát párjával rendelkező felhasználó tudja kinyitni és az objektumot csak a kinyitott szimmetrikus kulccsal tudja kikódolni.

Természetesen több ilyen módosító objektum is kapcsolható egy adott objektumhoz, azaz a jogosultságok adása ezekkel történik. A megfelelő jogosultsággal rendelkező felhasználó a működés során bármikor oszthat ilyen jogosultságokat. Ebben az esetben a módosító objektumnak köszönhetően az is visszakereshető, hogy ki osztotta az adott jogosultságot.

A titkos kulcsok kompromittálódása esetén a felhasználóknak új kulcspárra van szükségük. E mellett a felhasználóknak lehetőségük van a régebbi kulcspárok megtartására is, ugyanis a rendszer egy kulcsforgatás alkalmával nem kódolja át a szimmetrikus kulcsot tartalmazó módosító objektumokat.

## ***A workflow rendszer***

Egy workflow egy adott munkafolyamat leírása. A munkafolyamatot elemi feladatokból álló irányított gráffal ábrázolja. A folyamat leírásában lehetnek elágazások, mely után több ágon folyik a működés, az előrehaladást meghatározó feltételek, automatikusan elvégezhető feladatok. Az egyes feladatokhoz jogosultságok és határidők adhatók meg, valamint más objektumokra mutató hivatkozások adhatók meg hozzá, ezzel is jelezve, hogy az adott lépésben mely objektummal van feladata a felhasználónak. A rendszerben a workflow is egy objektum típusként jelenik meg és bárhol az objektum hierarchiában létrehozható valamely típusból egy új példány.

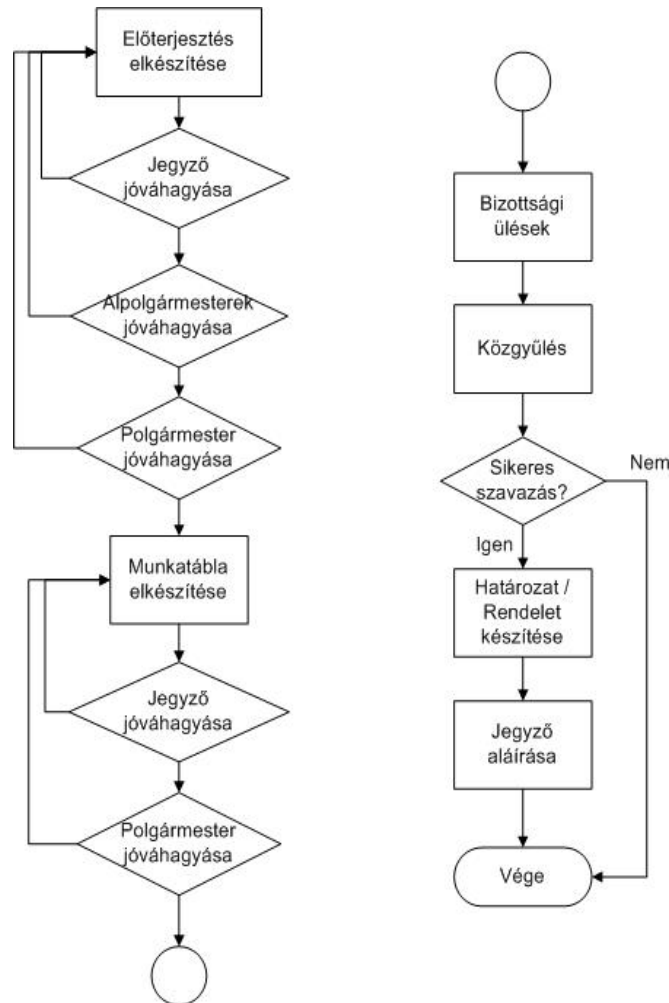
A workflow rendszer megvalósításának legnagyobb nehézsége a központi szerver hiánya volt. Ennek hiányában az egyes csomópontokon futó workflow rendszereknek konzisztens állapotban kell maradniuk, ugyanazt a működési utat kell bejárniuk és ugyanazokra a döntésekre kell jutniuk.

Mivel a objektumok szinkronizálásáról a tár gondoskodik ezért a workflow rendszer is tárban lévő objektumokban tárolja az állapotát. Egy workflow állapotát az adja meg, hogy az indulása óta milyen utat járt be, így a workflow módosító objektuma egy lépést ábrázol. Összekeveredés vagy késés esetére ez a módosító objektum tartalmazza az összes korábbi lépést is.

A központi szerver hiányát pedig úgy kerüli ki a rendszer, hogy minden workflownak van egy koordinátora, aki az új módosító objektumokat létrehozza és a tárba teszi, mégpedig az a csomópont amelyiken létrehozták az adott workflowt.

### **Elektronikus közgyűlési rendszer**

A rendszert bemutatandó elkészítettünk egy elektronikus közgyűlési rendszert és elkészítettünk workflowt, amely végigvezeti a felhasználókat a közgyűlés előkészületeitől egész az utómunkálatokig. Az egyszerűsített közgyűlés működése a következő:



3. ábra, a közgyűlés folyamata

A testületi referens összegyűjti az igazgatóságokról azok előterjesztéseiket, amelyek alapján összeállít egy munkatáblát.

A munkatáblának ezután egy jóváhagyási procedúrán kell átmennie, amely során a jegyzőnek, az alpolgármestereknek és a polgármesternek kell elfogadnia. Ezek közül bármelyik visszautasíthatja a munkatáblát, ekkor az indoklással visszakerül a testületi referenshez, aki átdolgozza.

Ez után a testületi referens elkészít az előterjesztésekhez egy határozati javaslatot, amit szintén jóváhagyat.

Majd az előterjesztések és a javaslatok a bizottságokhoz kerülnek, akik az üléseiken megtárgyalják azokat és véleményezik.

A közgyűlésben ismét megvitatják a véleményezett dokumentumokat és szavaznak az elfogadásukról és jegyzőkönyvet készítenek az ülésről.

Sikeres szavazás esetén a testületi referens a határozatból egy rendeletet készít, amit a jegyzővel aláírat és publikálás céljából a megfelelő igazgatóságra továbbít.

A közgyűlési rendszer workflow ábrája a 3. ábrán látható.

## **Összefoglalás**

A Peer2Meet projekt célja egy olyan elosztott architektúra elkészítése volt, amely könnyen felhasználható laikus felhasználók számára készített csoportmunka rendszerekhez. Ezzel egy új alternatívát kíván nyújtani a jelenleg is rendelkezésre álló dokumentumkezelő és csoportmunka rendszerek mellett.

A rendszer további előnye, hogy egy workflow kezelő rendszer is bele van integrálva, amely képes a felhasználók munkájának vezetésére, megkönnyítésére. Ezt a workflow rendszert felhasználva elkészítettünk egy önkormányzatoknak szánt elektronikus közgyűlési rendszert, amely kiválóan alkalmas az architektúra új lehetőségeinek a bemutatására.

## **Abstract**

The aim of Peer2Meet project is to provide a base for distributed groupware series and to build an electronic meeting support system. It is based on a peer2peer package and it implements a distributed object store. The system passes the objects stored in the object store to the other nodes using peer to peer protocol. Objects are encrypted and the right persons can only access them. The necessary object types for a meeting support system are for example: document, order, voting, message board, etc. A workflow system was implemented in this distributed environment. There is a workflow manager instance running on every node and nodes keep them synchronized by messages.