

Програм хангамжийн системүүдийн нэгтгэл

Програм хангамжийн архитектур (Software Architecture)

2012

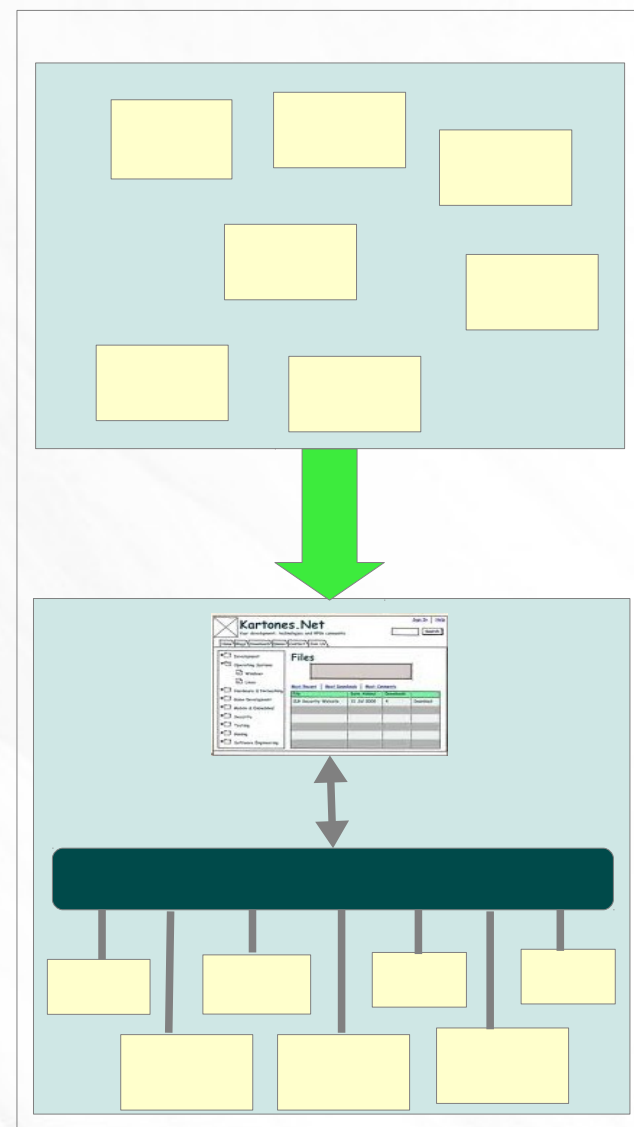
С. Бадрал

Агуулга

- Хэрэгцээ
- Файл дамжуулалт (File transfer)
- Хамтран хэрэглэх өгөгдлийн сан (Shared DB)
- Алсын дуудалт (RMI)
- CORBA
- Мессеж (Messaging)
- Дүгнэлт

Нэгтгэлийн учир шалгаан

- Хэрэглэгч тархан суурилсан хэд хэдэн програм хангамжийн функцүүдийг дуудах хэрэгцээтэй бол
- Үүний тулд тус тусдаа буй программууд нэгтгэгдэн холбоотой ажиллах ёстой.
- Гэхдээ
 - Сүлжээ удаан
 - Сүлжээ найдваргүй
 - Програмууд янз бүр
 - Өөрчлөлт хийхээс зайлсхийх аргагүй



Нэгтгэл хийх үндсэн дөрвөн арга зам

- File transfer (Файл дамжуулалт)

Нэг програм нь нөгөө програмд ашиглагдах өгөгдөл агуулсан файлыг үүсгэх буюу нөгөө талаас нөгөөгийн үүсгэсэн файлыг уншина.

- Shared Database (хамтран хэрэглэх өгөгдлийн сан)

Янз бүрийн программууд өгөгдлөө хамтран хадгалж мөн хэрэглэнэ.

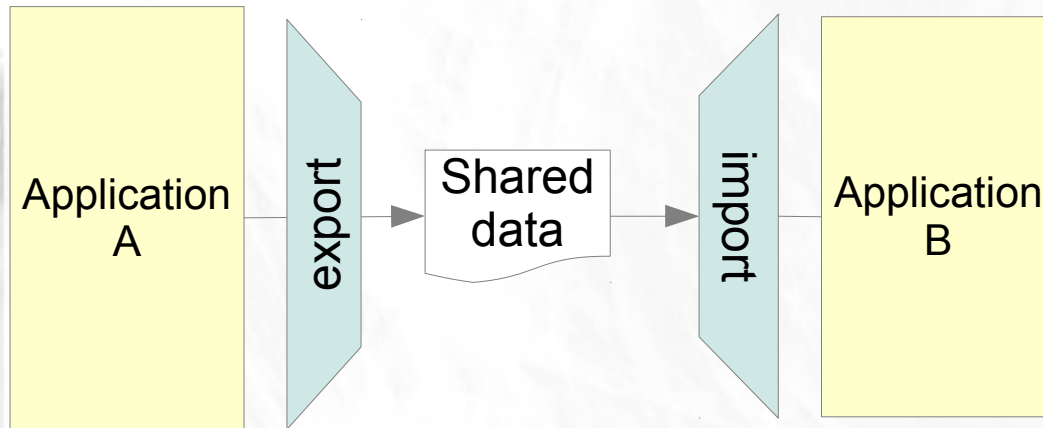
- Remote Method Invocation (алсын дуудалт)

Програм тусбүр өгөгдөл солилцохын тулд өөр програмд ашиглагдаж алсын дуудалт хийх функцийн хэсгээ бэлтгэнэ.

- Messaging (мессеж)

Бүх програм нэг ерөнхий мэдээний системд холбогдоно. Мэдээний тусламжтай өгөгдлөө солилцож, функүүдээ дуудаж хэрэглэнэ.

File transfer



Урьдач нөхцөл

- Нэг файлын системтэй байх
- Форматуудаа тогтсон байх (текст, XML, CSV, etc.)
- Илгээгч болон хүлээн авагч форматаа нэгэн ижил ойлгож байх

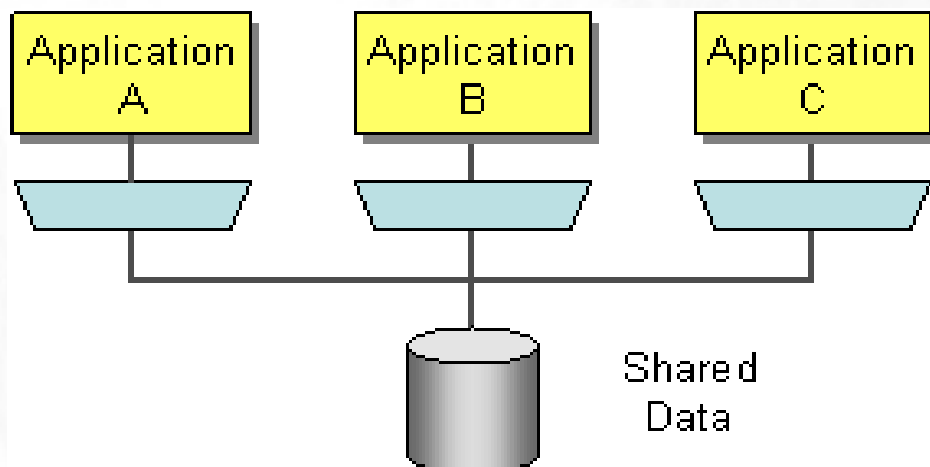
Давуу тал

- Өөр програм хэрэглэхгүйгээр хялбархан хэрэгжүүлэх боломжтой
- Бүх платформ, програмчлалын хэлийн хувьд боломжтой
- Асинхрон, програмууд тус тусдаа цааш ажиллах боломжтой
- Програмуудын хооронд бага хэрээс
- Зөвхөн экспорт хийх зурвас
- Програмын дотоод бүтэц далд хэвээр үлдэнэ

Сул тал

- Өгөгдөл шинэчлэх хугацааны интервал урт тул авцалдаагүй байдал үүснэ.
- Хөгжүүлэгчид эсвэл хэрэглэгчид хооронд баахан дүрэм тогтох шаардлагатай.
- Хавтас, файлын нэр, үүсгэх болон устгах хугацаа
- Өгөгдлийн симантек програмуудад зөвхөн локалаар тодорхойлогдоно.

Shared database



Урьдач нөхцөл

- Програмууд нэг ерөнхий ө.с-г хамтран хэрэглэх боломжтой байх
- Өгөгдлийн нэг ерөнхий схемтэй байх

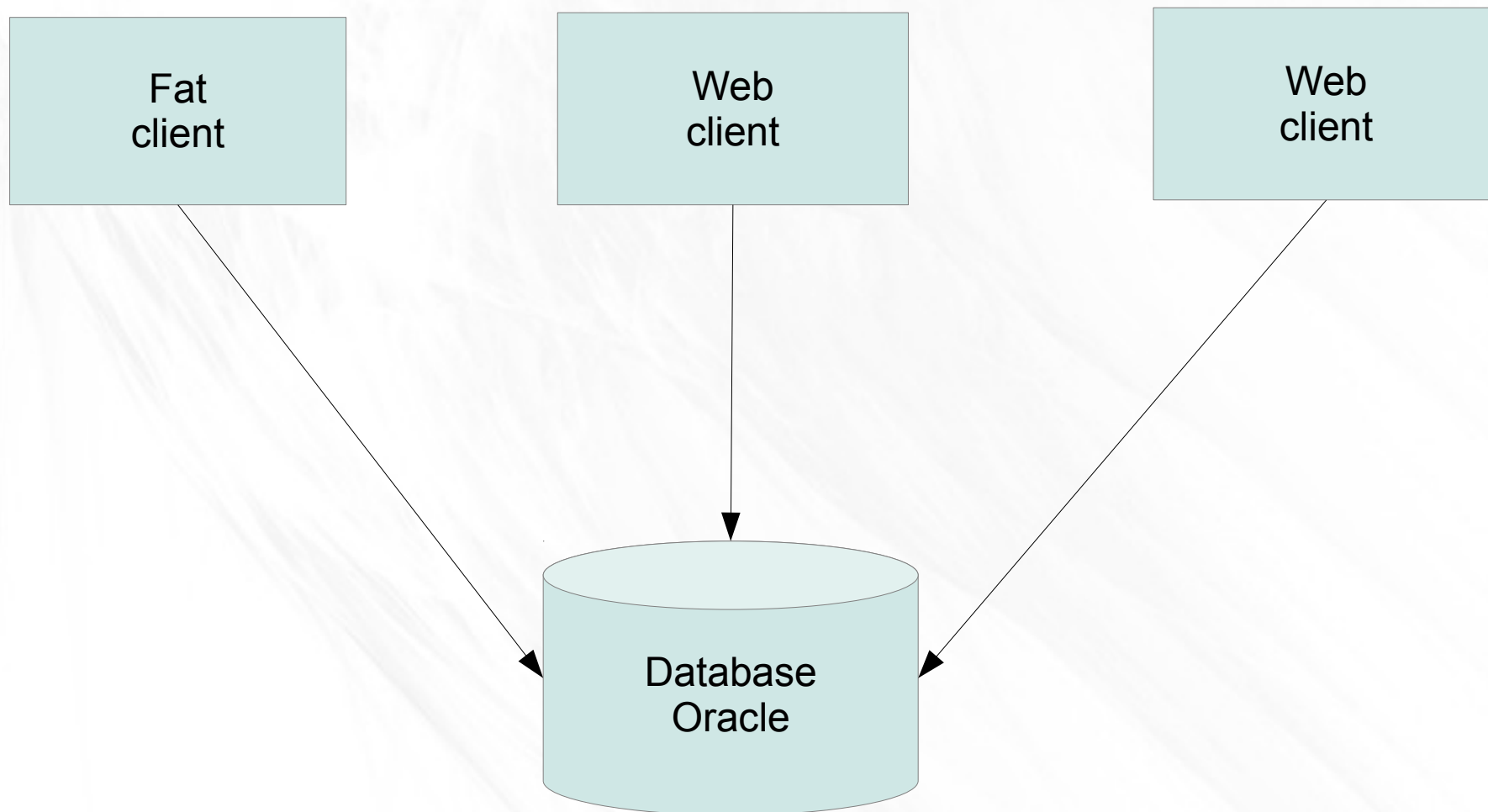
Давуу тал

- Зэрэг хандалтын хувьд транзакшн менежменттэй шууд шинэчлэгдэнэ
- SQL / OR буулгалтууд өргөн тархсан ба өөр програм шаардлагагүй
- Хамтарсан бүх програмуудын өгөгдөл авцалдаатай, синхрон, симантик зөрчил бага
- Сайн өгөгдлийн загварчлал хийх шаардлага

Сул тал

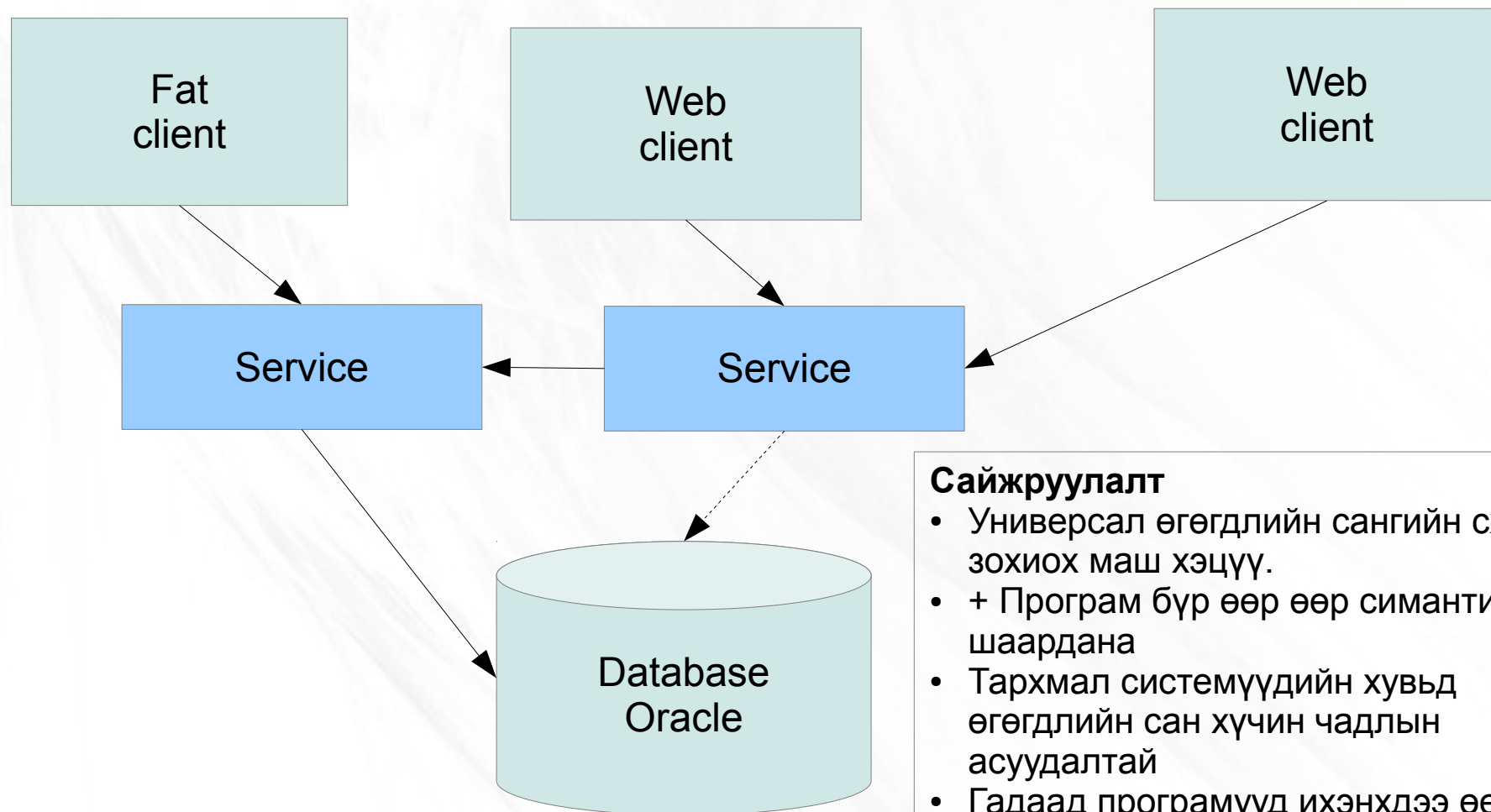
- Универсал өгөгдлийн сангийн схем зохиох маш хэцүү.
- Програм бүр өөр өөр симантик шаардана
- Тархмал системүүдийн хувьд өгөгдлийн сан хүчин чадлын асуудалтай
- Гадаад програмууд ихэнхдээ өөрийн Ө.С.тай байдаг
- Өгөгдөл хайрцаглалтгүй солилцоно
- Өндөр хэрээс: Өгөгдлийн сангийн схемд хийсэн өөрчлөлт бүх програмд нөлөөлнө

Жишээ



Үйл ажиллагаа бүх клиентэд давхацна, өөр шийдэл бий юу?

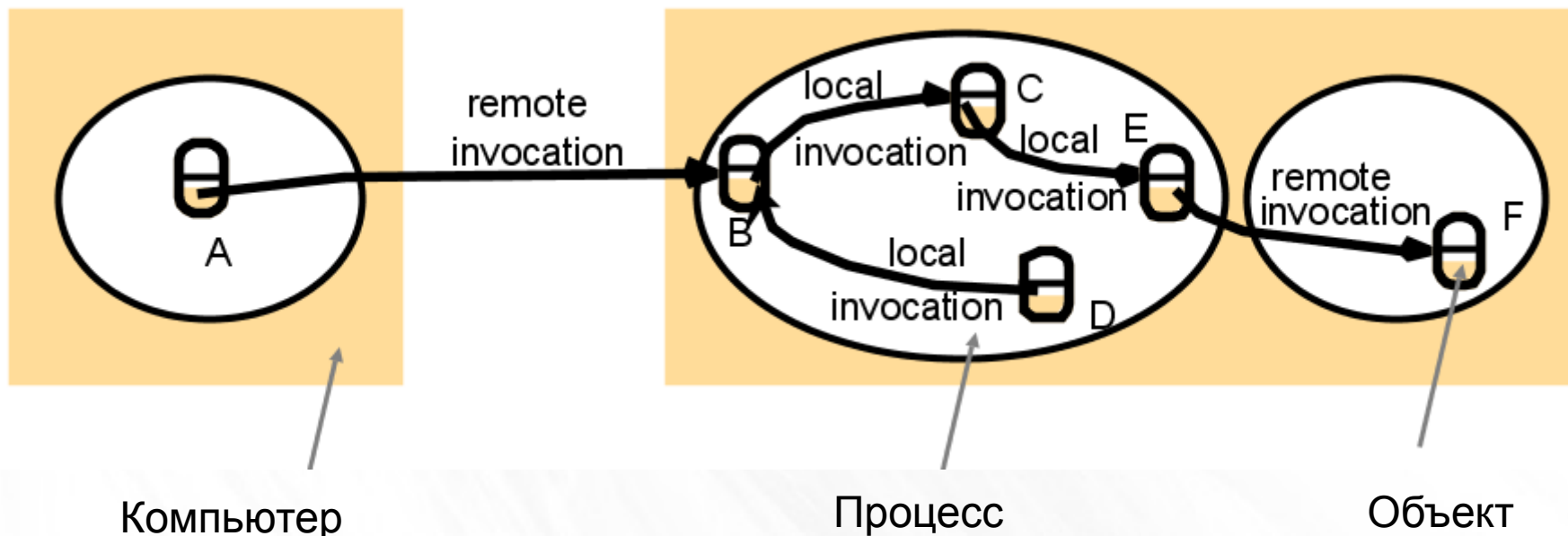
Жишээ – Сайжруулалт - SOA



Сайжруулалт

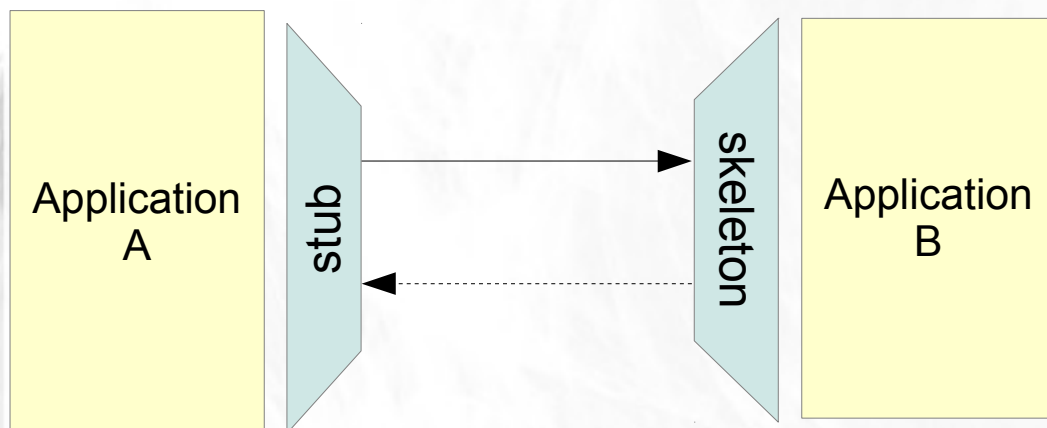
- Универсал өгөгдлийн сангийн схем зохиох маш хэцүү.
- + Програм бүр өөр өөр симантик шаардана
- Тархмал системүүдийн хувьд өгөгдлийн сан хүчин чадлын асуудалтай
- Гадаад програмууд ихэнхдээ өөрийн Ө.С.тай байдаг
- + Өгөгдөл хайрцаглалтгүй солилцоно
- + Өндөр хэрээс: Өгөгдлийн сангийн схемд хийсэн өөрчлөлт бүх програмд нөлөөлнө

Remote Method Invocation



- Алсын объектыг хэрхэн дуудах алсын зурвасыг тодорхойлно
Тодорхойлолт
 - Зурвасын нэр
 - Өгөгдлийн төрлийн тодорхойлолт
 - Алсын методуудын сигнатур
 - Методуудын нэр
 - Оролт гаралтын параметер
 - Буцаах утга
- MiddleWare бүр энэхүү зурвасыг тодорхойлох хэлийг агуулах ба IDL гэдэг.
- Marshalling ↔ Unmarshalling

Remote Method Invocation



Урьдач нөхцөл

- Өөр програмд хэрэгтэй функцийг хэрэгжүүлсэн байх
- Нэгдсэн технологи (CORBA, COM, .NET, RMI, SOAP with HTML)
- Сүлжээ тасрах, дуудсан програм гацахыг тооцоолж програмчлах

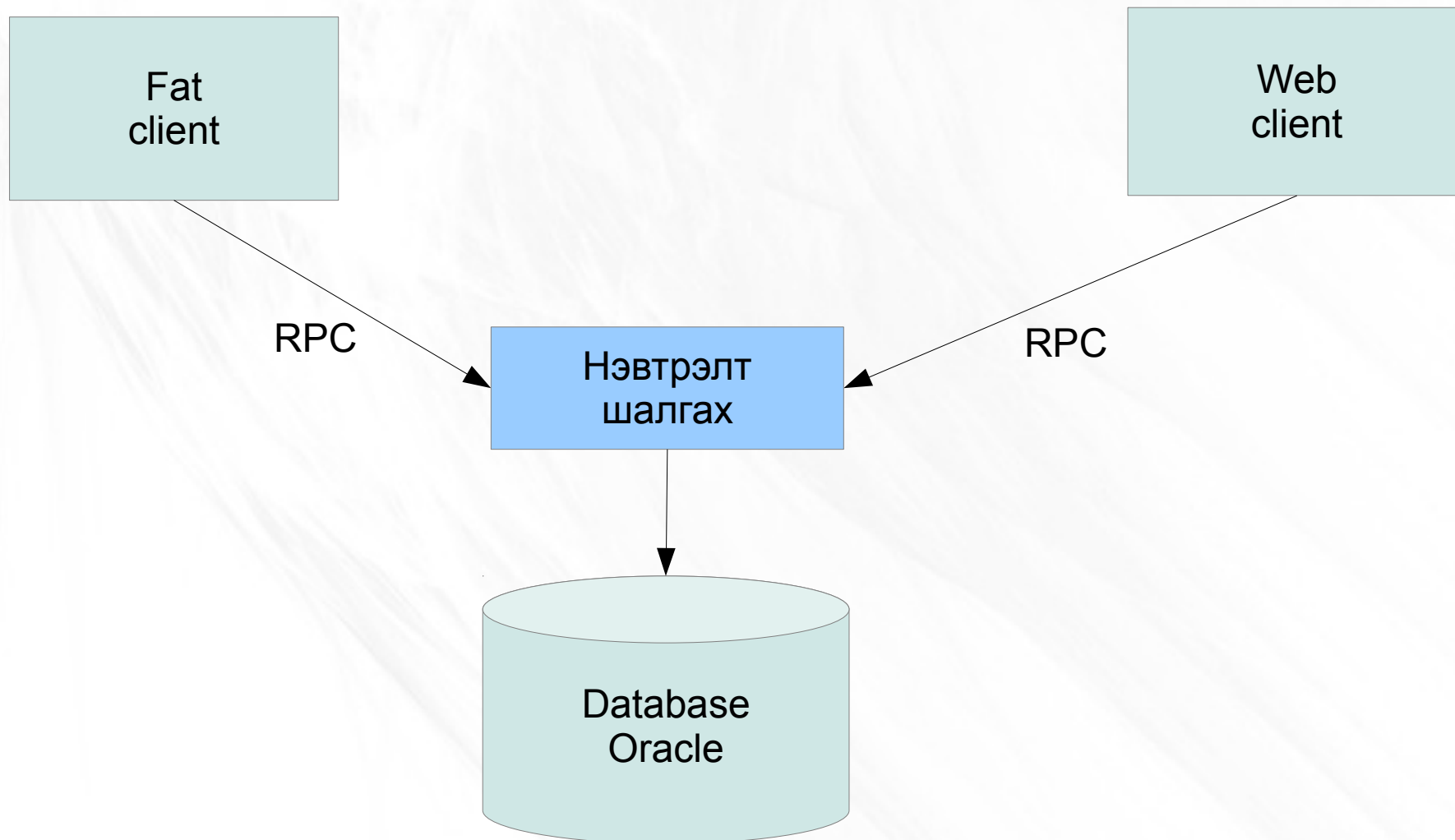
Давуу тал

- Функцийн дахин хэрэглээ
- Өгөгдөл нь локалаар хайрцаглагдсан тул програм бүр өөрийн өгөгдөлдөө дурын формат хэрэглэж болно
- Функц дахин хэрэглээ өндөр тул кодын давхцал бага

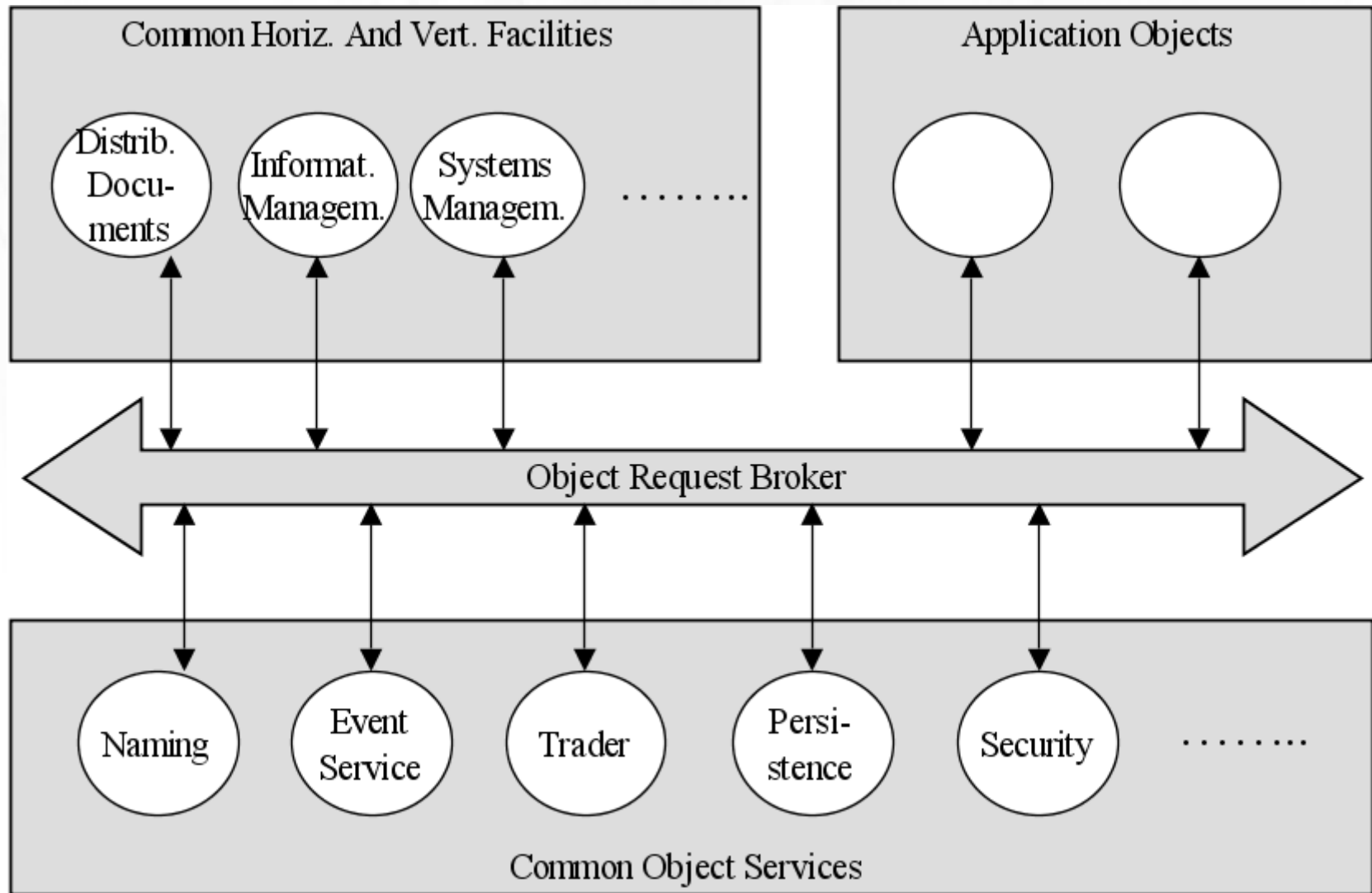
Сул тал

- Програм гацахад тогтвор муу бөгөөд хэрээс их
- Чадал ба гацалт тасралтыг тооцоолох ёстой
- Дуудагдаж буй програмд өөрчлөлт ороход дуудагч бүх програмд нөлөөлнө.
-

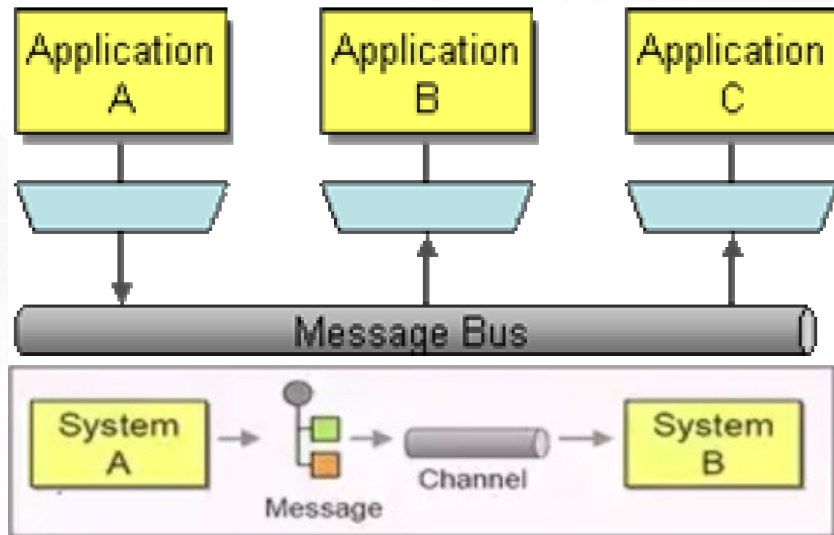
Жишээ: Стандарт сервисүүд



CORBA



Messaging



Урьдач нөхцөл

- Програмууд ямар нэг Message Bus-т холбогдсон байна
- Тохирох технологийг хэрэглэнэ. (JMS, IBM WebsphereMQ, Microsoft MSMQ, MQ Series, Asyncron Web services)

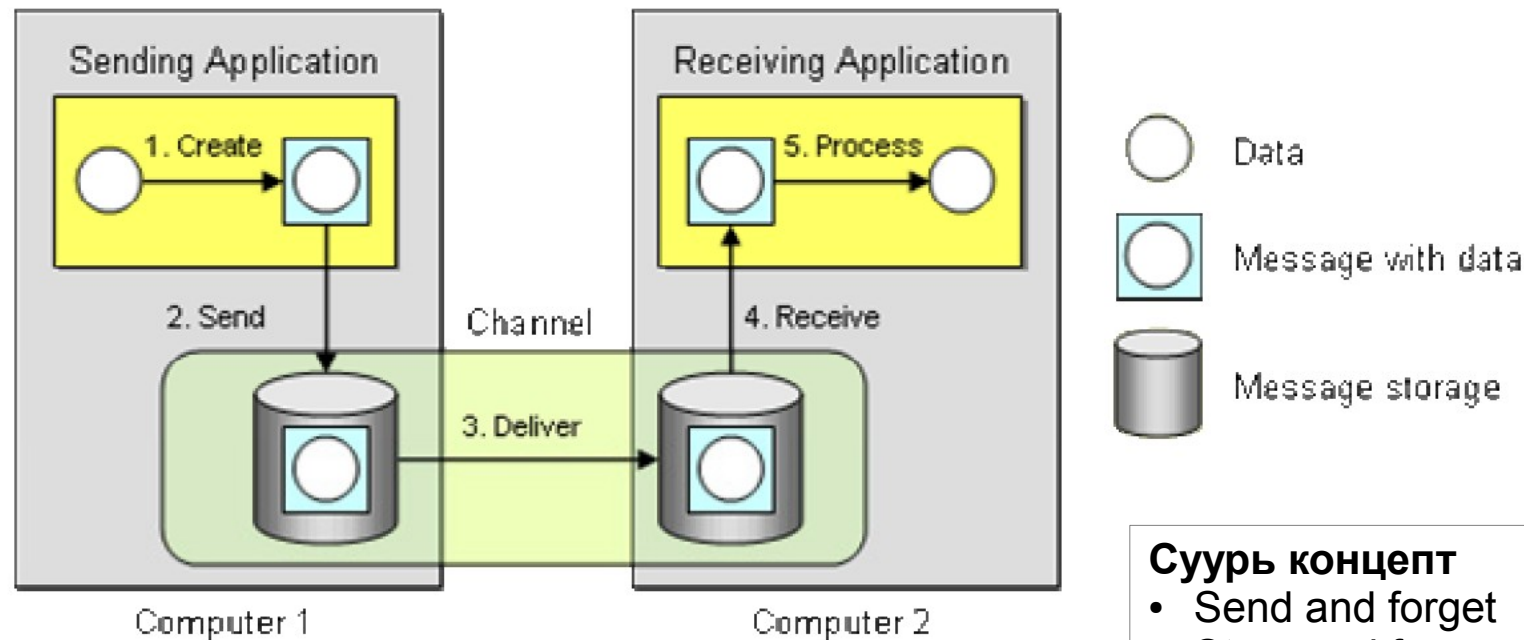
Давуу тал

- Бага хэрээс, бараг файл дамжуулалт шиг
- Нийт системийн хувьд нуралт гарах нь бага, учир нь мессежийг хадгалах боломжтой
- Мессеж дараалалтай тул чадлын асуудал гарах нь бага
- Асинхрон харилцаа хугацааны хүлээлттэй боловч мэдэгдэнэ.
- Мессежээр өгөгдөл дамжина.

Сул тал

- Асинхрон харилцааг хийхэд хэцүү
- Програм хоорондын симантик зөрчилдөөний асуудал шийдэгдээгүй

Message-oriented Middleware (MOM)



Суурь концепт

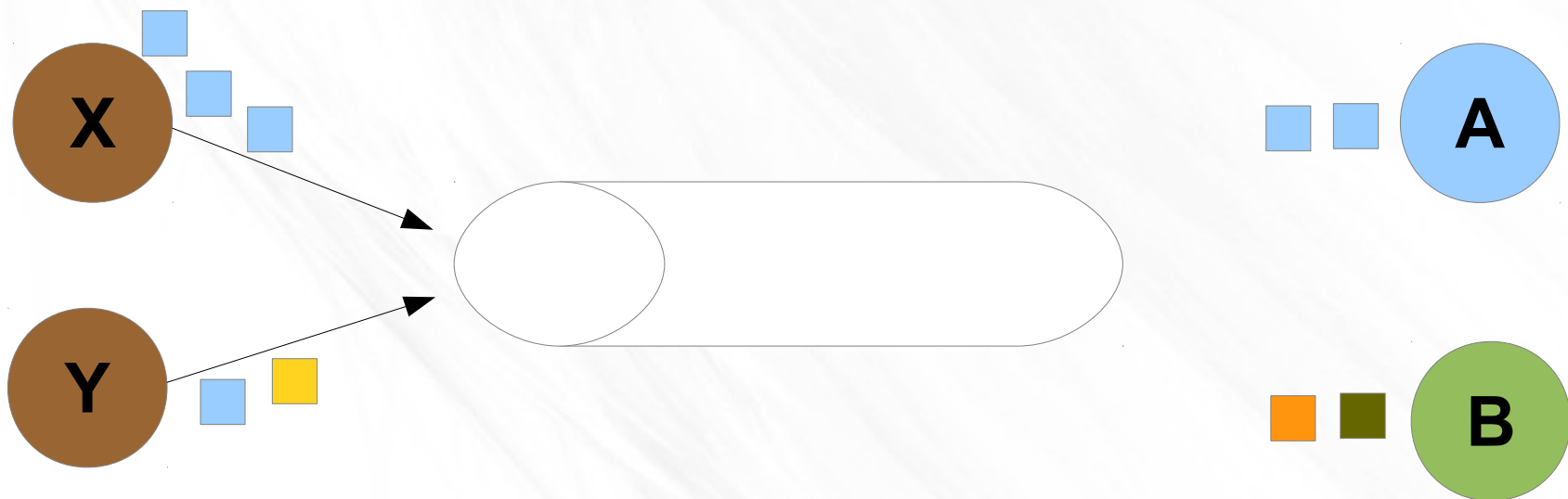
- Send and forget
- Store and forward.

Бага хэрээс ба найдвартай байдал

- Channel-ууд программаас тусгаарлагдсан → Байрлалын хамаарал үгүй
- Channel-ууд асинхрон бөгөөд найдвартай → Түр хамаарал байхгүй
- Өгөгдөл мессеж дотор хайрцаглагдсан → Өгөгдлийн форматын хамаарал үгүй

МОМ-н нийлүүлэлт

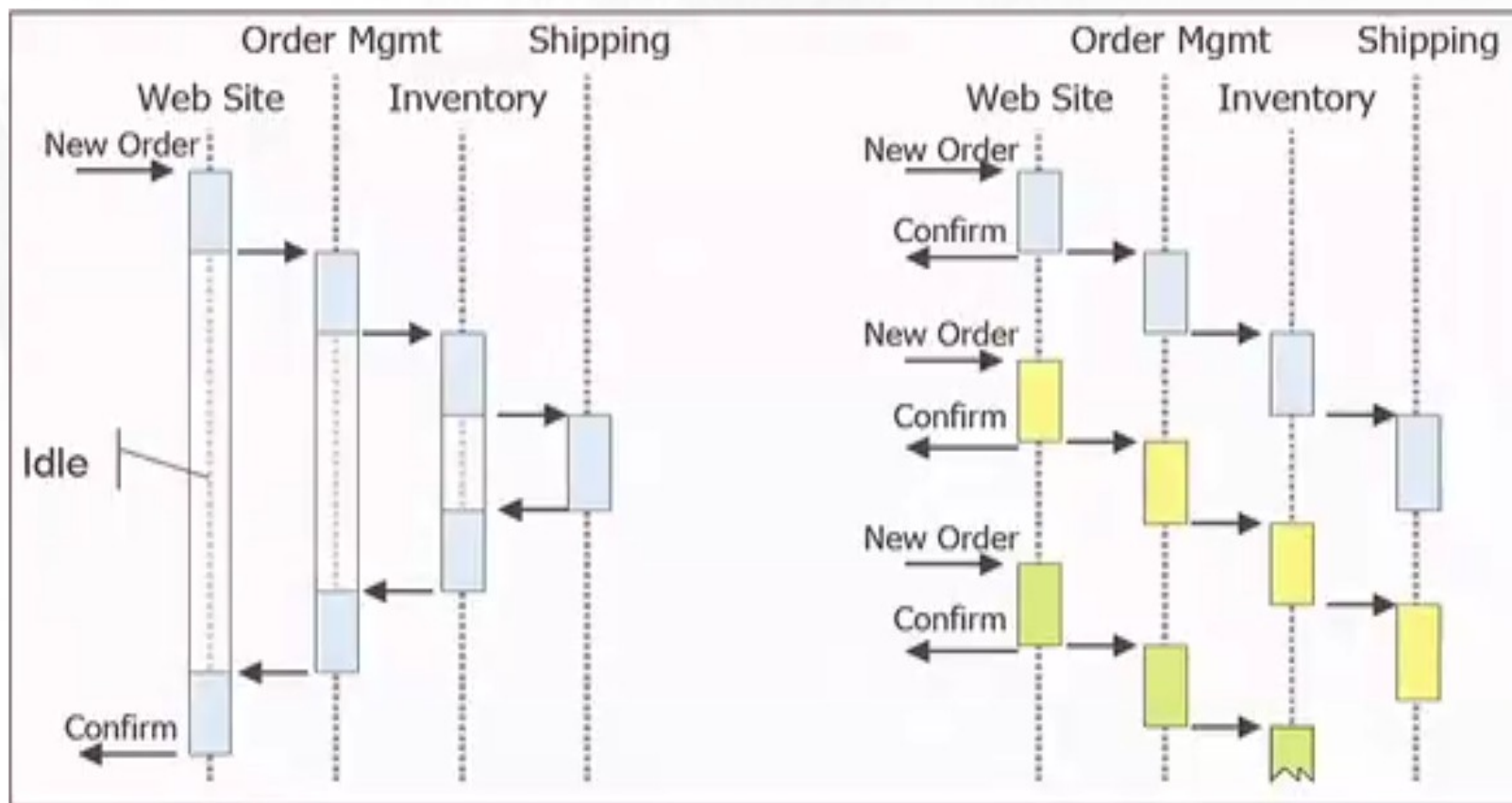
- Мессежүүд сервер дээрх дараалалд (Queue) хадгалагдана.
- Нэг үйлчүүлэгчээр (consumer) зөвхөн нэг удаа л уншигдана
- Олон нийлүүлэгч / үйлчлүүлэгч (Producer/Consumer) тухайн дарааллаа (queue) л хэрэглэнэ.



MOM-н шаардлага

- Төвөгтэй програмчлалын загвар
 - Event driven програмчлалын загвар
 - Eventhandler дээр логик тарсан
 - Процессын хязгаараар debug хийгддэг
- Мессеж дараарлал
 - Мессеж нь нийлүүлэх баталгаатай боловч дарааллын баталгаагүй
 - Хараат мессежүүд дахин жагсах ёстой
- Синхрон хэрэглээ
 - Хэрэглэгч хүсэлтээ синхрон боловсруулалт байна уу үгүй юу хамаагүй
 - Программ шууд хариулж байх ёстой
- Хүчин чадал
 - Их хэмжээний өгөгдөл MOM-с гадуур бэлтгэгдэх (initialization) нь дээр
- Платтформ дэмжилт
 - MOM эсвэл платтформ боломжийг хязгаарладаг
- Үйлдвэрлэгчээс хамаарна
 - JMS гэх мэт стандарт бий боловч хэрэгжүүлэлт үргэлж платтформоос хамаардаг
 - Нэгтгэсэн шийдлийн нэгтгэлд цааш шаардлага тулгарна

Synchron vs. Asynchron



Асинхрон харилцааны нөлөөлөл

Процессууд зэрэг ажиллана → Хурдан боловсруулалт, төвөгтэй debug

Үр дүн CallBack-р ирдэг → Шинэ бичвэр дахь хүлээн авалт ба боловсруулалт өөр ажил гүйцэтгэнэ

Тодорхойгүй гүйцэтгэлийн дараалалтай → Синхрончлолын логигоос хамааралгүй процессууд

Commercial messaging systems

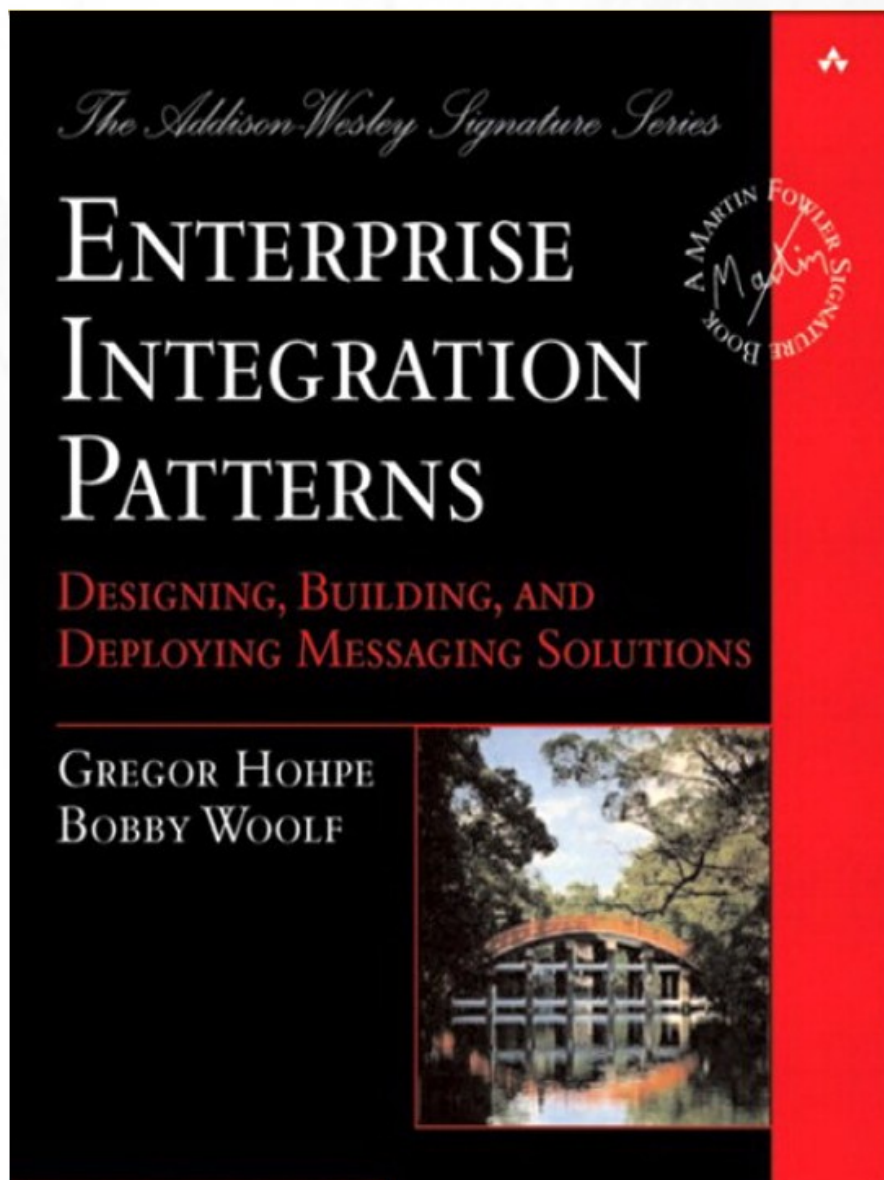
- Үйлдлийн системүүд
Messaging үйлдлийн системүүдэд улам ихээр хэрэглэгдэж байна
 - MSMQ нь Виндовс 2000/XP/2008Server/Vista/7/8, Oracle AQ нь Oracle-н нэг хэсэг
- Программын сервер (Application server)
 - J2EE нь JMS тодорхойлсон, SUN J2EE JDK-дээ хэрэгжүүлсэн
 - J2EE програмын сервер JMS-г мөн хэрэгжүүлсэн (Ж.нь IBM Websphere, BEA Weblogic)
- EAI suites
 - Төлбөртэй
 - IBM Websphere MQ, Microsoft BizTalk
- Web service toolkits
 - Вэб сервисээр стандарт messaging
 - Вэб сервист суурилсан янз бүрийг шийдлүүд

Бүгд өөр нэртэй боловч төстэй даалгаврыг гүйцэтгэнэ.

Commercial messaging systems

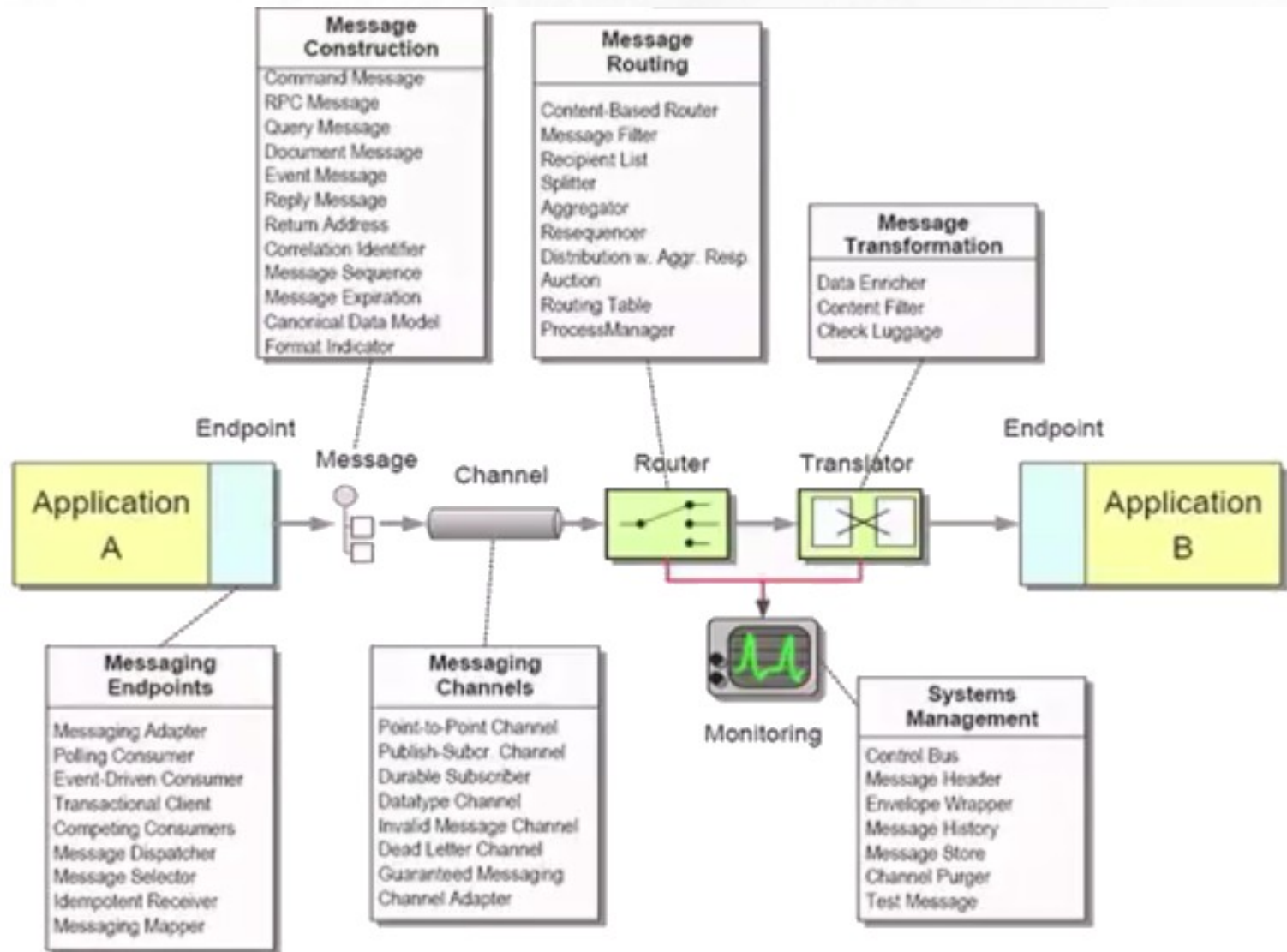
Enterprise Integration Patterns	Java Message Service (JMS)	Microsoft MSMQ	WebSphere MQ	
Message Channel	Destination	MessageQueue	Queue	
Point-to-Point Channel	Queue	MessageQueue	Queue	
Publish-Subscribe Channel	Topic	---	---	
Message	Message	Message	Message	
Message Endpoint	MessageProducer, MessageConsumer			
Enterprise Integration Patterns	TIBCO	WebMethods	SeeBeyond	Vitria
Message Channel	Topic		Intelligent Queue	Channel
Point-to-Point Channel	Distributed Queue		Intelligent Queue	Channel
Publish-Subscribe Channel	Subject	---	Intelligent Queue	Pub/Sub Channel
Message	Message	Document	Event	Event
Message Endpoint	Publisher, Subscriber	Publisher, Subscriber	Publisher, Subscriber	Publisher, Subscriber

65 паттерн [Gregor, Hohpe, Bobby Woolf]



1. Request-Reply жишээ
2. Order Management
жишээ
3. Bonus жишээ

65 паттерн [Gregor, Hohpe, Bobby Woolf]



Дүгнэлт

- Нэгтгэлийн аргачлалууд
 - Файл дамжуулалт
 - Remote Procedure Invocation
 - CORBA
 - Shared database
 - Messaging
- Архитектурч болон програмистуудад тавигдаж буй шинэ шаардлага
- Hore/Woolf -н бүтээх аргачлалууд