Resource-Oriented Computing 그리고 IT의 미래

신재호 <netj@sparcs.org>

2008-01-24

Google Korea

이번 시간...

- 목표
  - 느껴봅시다: "**ROC가 좋은 거구나, 필요하겠다**"
- 순서
  - 두 가지 **난제**
  - **ROC**가 도대체 뭔가
  - ROC의 **예**
  - ROC로 생기는 **좋은 일**들

“효율”에 관한 두 가지 난제

- Reuse
  - 잘 만들자
  - 추상화
- Optimization
  - 잘 돌리자
  - 구체화

Reuse?

- Software Engineering의 꿈
  - **고치지 않으면서 바꾼다**
- Methodologies & Programming Paradigms
  - Structured, Object-oriented, Functional, Declarative Programming; Scripting, Domain Specific Languages; Component-based Development; …
  - ASM, C, Smalltalk, C++, Java, Eclipse, C#, .NET, LISP, ML, Haskell, Prolog, Perl, Python, Ruby, PHP, JSP, SQL, HTML, CSS, XSL, XQuery, UML, IDL, …

Optimization?

- System Engineering의 꿈
  - **더 가볍게, 더 빠르게, 상황에 꼭 맞게**
- Multi-core Processor 시대
  - Software는 더 이상 공짜로 빨라지지 않는다
  - 책임 이동: Hardware --> Software
- **Scalability**!
  - Concurrency 문제

Resource-Oriented Computing

1. **Resource** is an abstract set of **information**
2. Each resource may be identified by one or more **logical identifiers**
3. A logical identifier may be resolved within an information-context to obtain a **physical resource-representation**
4. Computation is the reification of a resource to a physical resource-representation
5. Resource representations are **immutable**
6. Transreption is the isomorphic lossless transformation of one physical resource-representation to another
7. **Computational results are resource** and are identified within the address space

- *ROC Fundamentals*, [ROC Intro]

Resource-Oriented Computing

한 마디로,

사방팔방 구석구석, 이름을 붙여두고 써먹어라

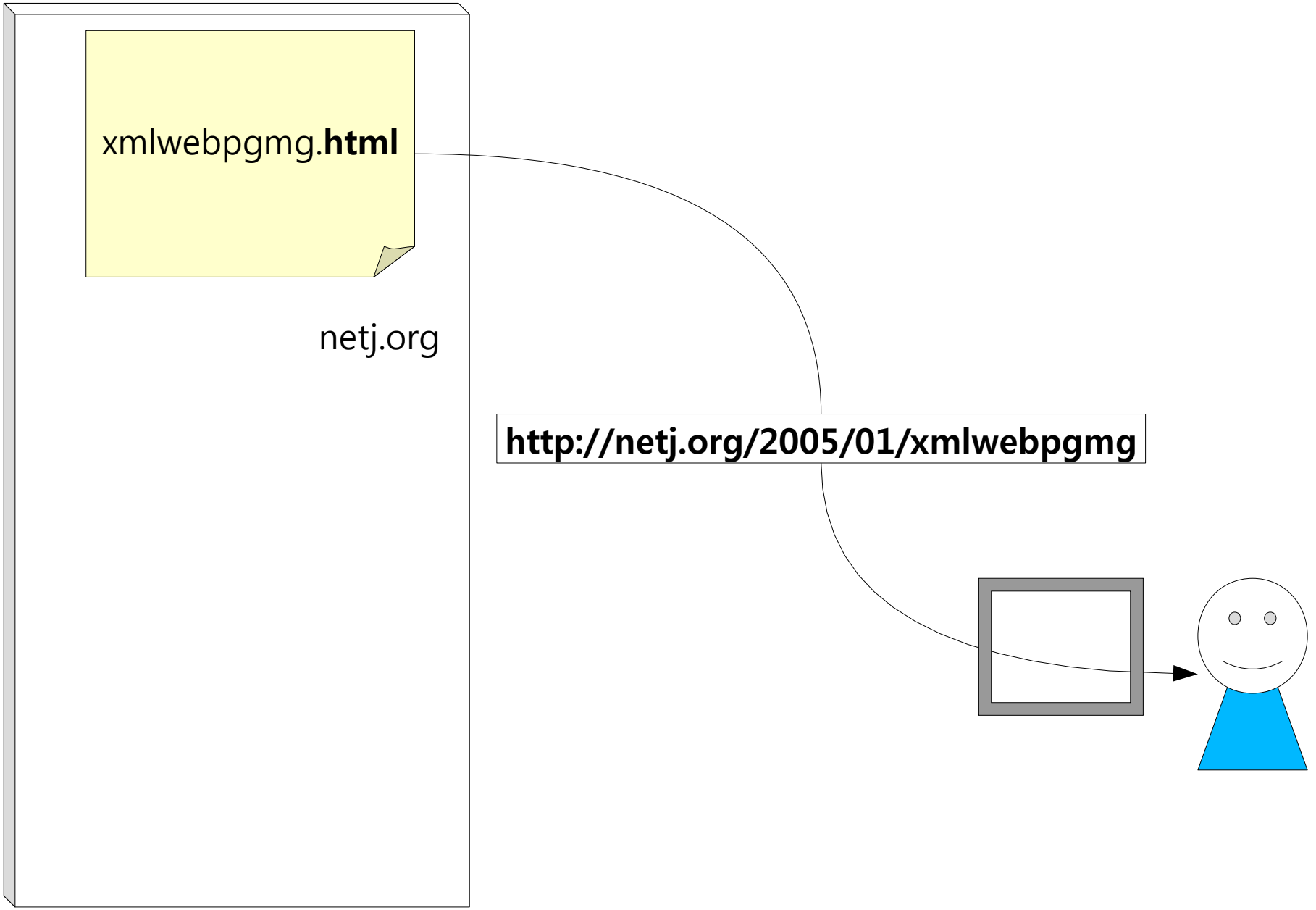
그럼 좋은 일들이 많이 생기더라

ROC의 예

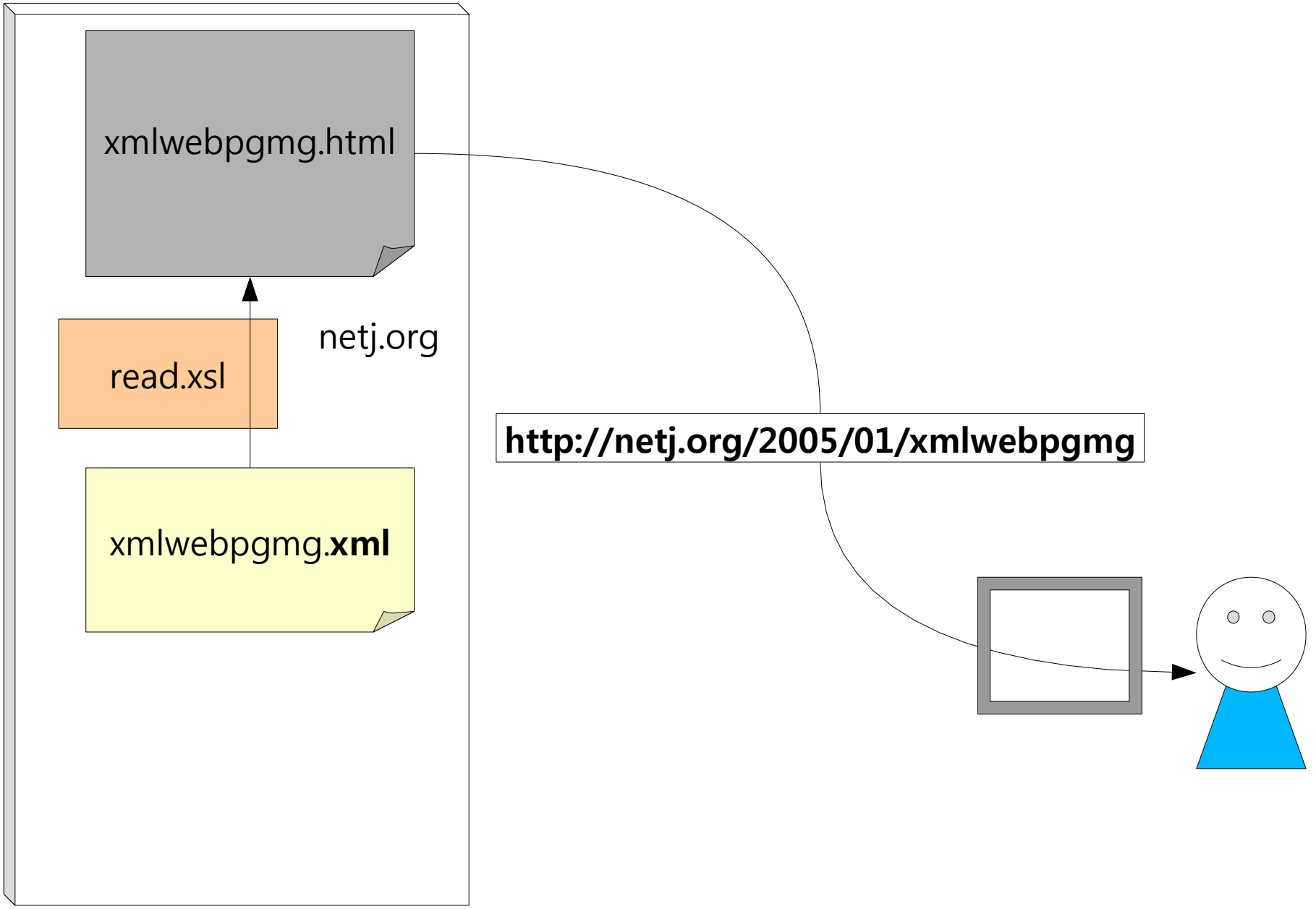
- Unix
  - 모든 게 파일
  - 경로가 id
  - |와 <, >로 연결
- Web
  - URL로 자료 접근
  - 링크로 연결

- NetKernel (1060 Research)
  - URI 기반
  - ROC를 위한 운영체제
  - 다양한 언어 지원
    - Java, Javascript, XSLT, XQuery, Python, Ruby, ...
  - 임의의 환경에 접속
    - HTTP, Swing, SMTP, ...

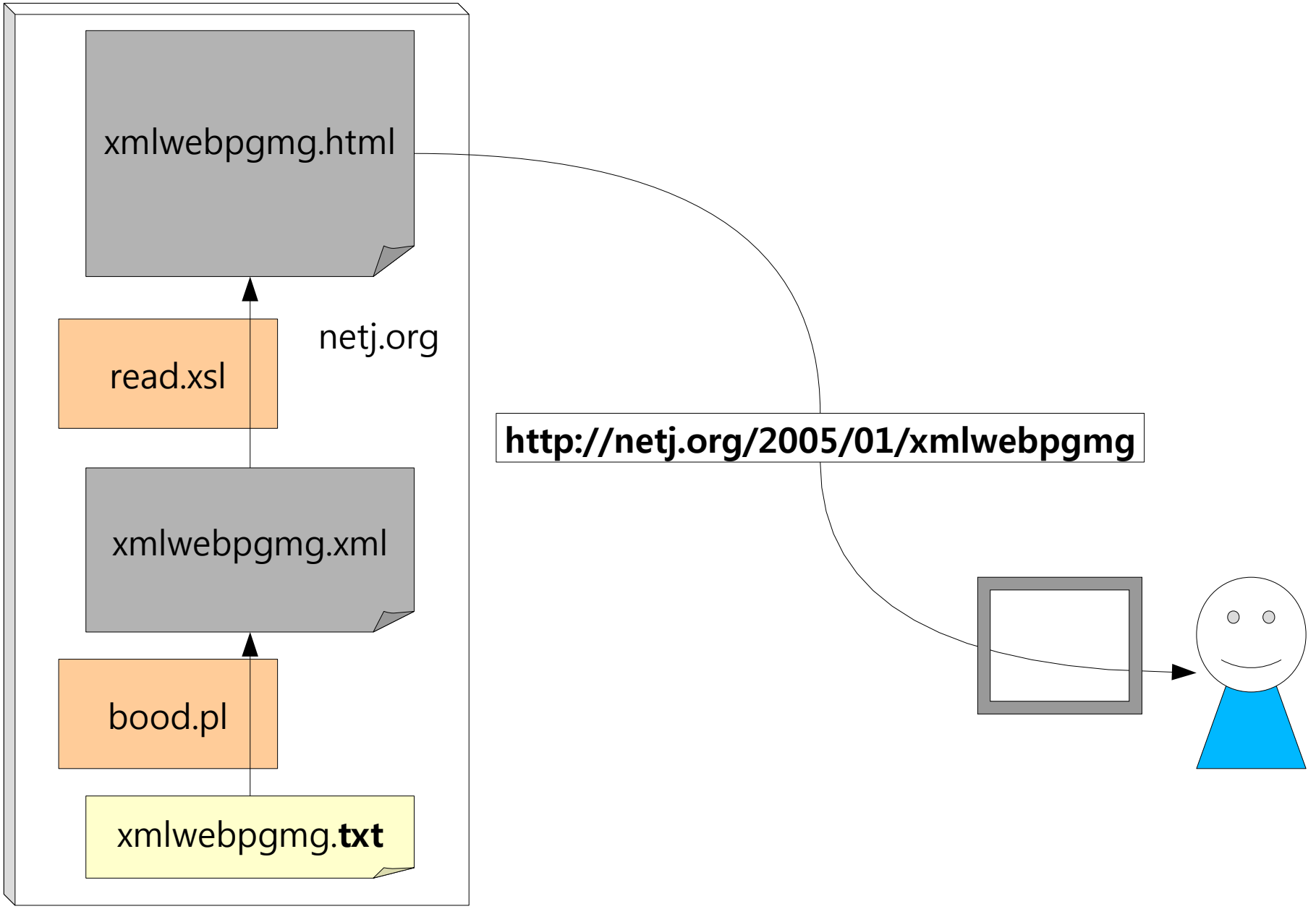
Web의 진화 1



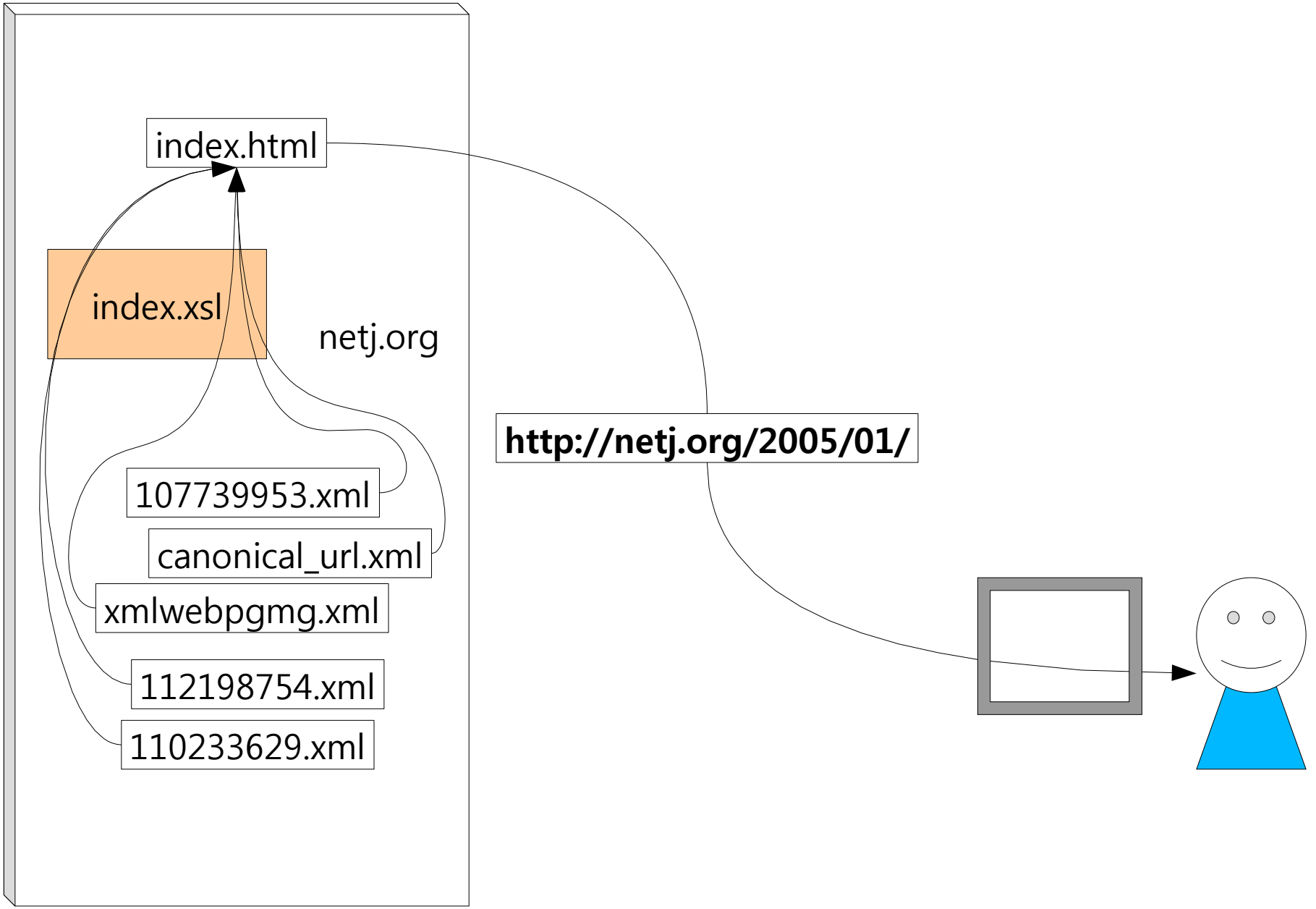
Web의 진화 2


xmlwebpgmg.html
netj.org
read.xsl
xmlwebpgmg.xml
http://netj.org/2005/01/xmlwebpgmg

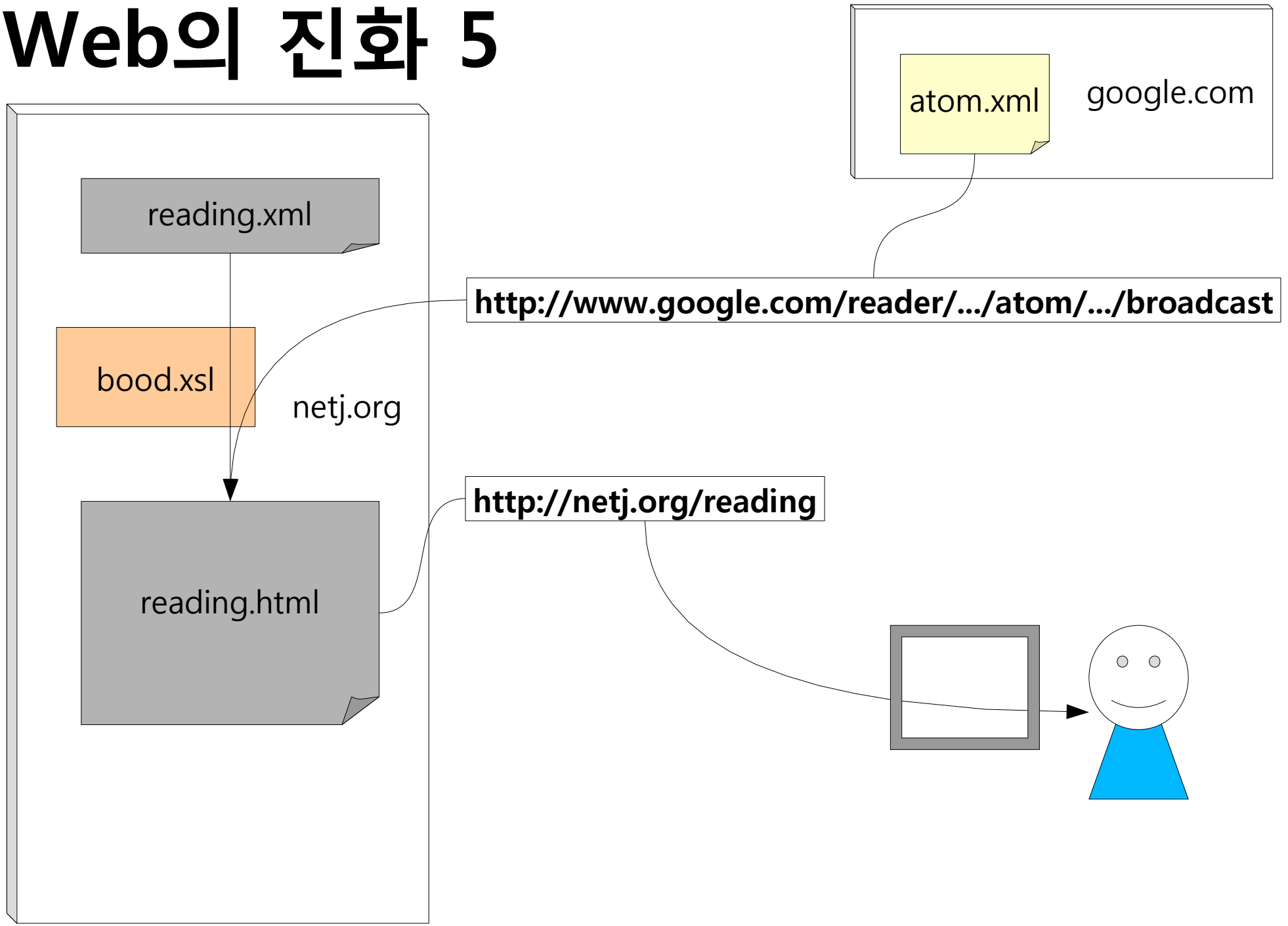
Web의 진화 3


xmlwebpgmg.html
netj.org
read.xsl
xmlwebpgmg.xml
bood.pl
xmlwebpgmg.txt
http://netj.org/2005/01/xmlwebpgmg

Web의 진화 4


index.html
index.xsl
netj.org
http://netj.org/2005/01/
107739953.xml
canonical_url.xml
xmlwebpgmg.xml
112198754.xml
110233629.xml

Web의 진화 5


atom.xml
google.com
reading.xml
http://www.google.com/reader/.../atom/.../broadcast
bood.xsl
netj.org
http://netj.org/reading
reading.html

Web의 진화와 ROC

- 안
  - CGI, PHP, ASP, JSP, Cocoon, Struts, ROR, ...
- 밖
  - Mashups
  - Open API

- ROC는 이 모두를 바르게 담아내는 이론
- NK는 이 모두를 통합해낼 수 있는 구현

ROC와 좋은 일들

- Reuse
- Optimization
- 소프트웨어 개발

ROC와 Reuse

- 통일된 이름 공간
- 통일된 인터페이스

- 가령, 이종의 Web Services를 Open API로 통합

ROC와 Optimization

- 캐시 기회
  - Id
  - Immutable resource
  - Reuse of computations
- 분산처리 기회
  - Explicit dependencies
  - Stateless computations [REST]
- 성능 자동적응
  - Computations on demand
  - Cache expiration based on resource value

ROC와 소프트웨어 개발

- 역할의 협력적인 분리
  - 개발자는 하나의 문제에만 집중할 수 있다
  - 설계자는 모든 권한을 쥐고 문제를 쪼갤 수 있다

- Construct, Compose, Constrain

정리

- Resource-Oriented Computing
  - 모든 것에 이름을 붙여두고 활용하자
  - 그러면 좋은 일들이 생긴다
    - Reuse와 Optimization 두 마리 토끼
    - 개발자와 설계자 모두의 행복

- ROC를 기반으로 IT의 행복한 미래를 다지자

참고자료

- [ROC Intro]
  - Introduction to Resource-Oriented Computing
  - http://www.1060research.com/resources/docs/IntroductionToResourceOrientedComputing-1.pdf
- NetKernel 

  - http://www.1060research.com/netkernel/
- [REST]
  - Representational State Transfer
  - http://en.wikipedia.org/wiki/Representational_State_Transfer