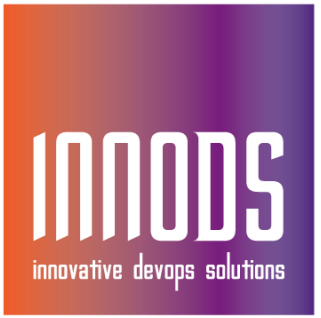


클라우드 환경에서 소프트웨어 배포를 위한 시스템 개발

공개 소프트웨어 신시장 창출 지원사업

클라우드 컴퓨팅 환경에서의 소프트웨어 배포 및 보안 분야

최종 발표

2012년 12월 14일

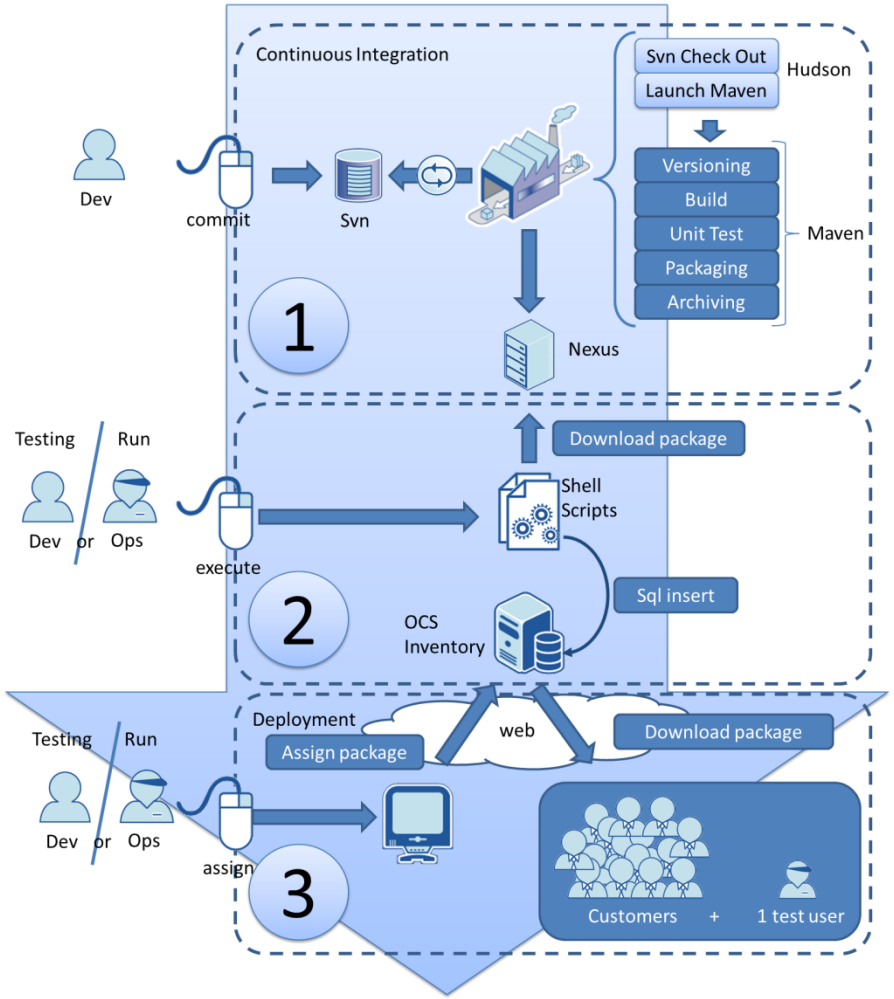
㈜이노디에스

개발 목표(1)

"WAS(Web Application Server)에서 동작하는 JVM(Java Virtual Machine) 기반 소프트웨어를 쉽고 빠르게 배포한다"



개발 목표(2): 타사 사례(C#/.Net 배포 자동화)



First click: a svn commit launches the Continuous Integration process

Second click: OCS package creation

Third click: assign the package to users

http://blog.octo.com/en/continuous-delivery-how-do-we-deliver-in-3-clicks-to-7000-machines/

개발 목표(3)

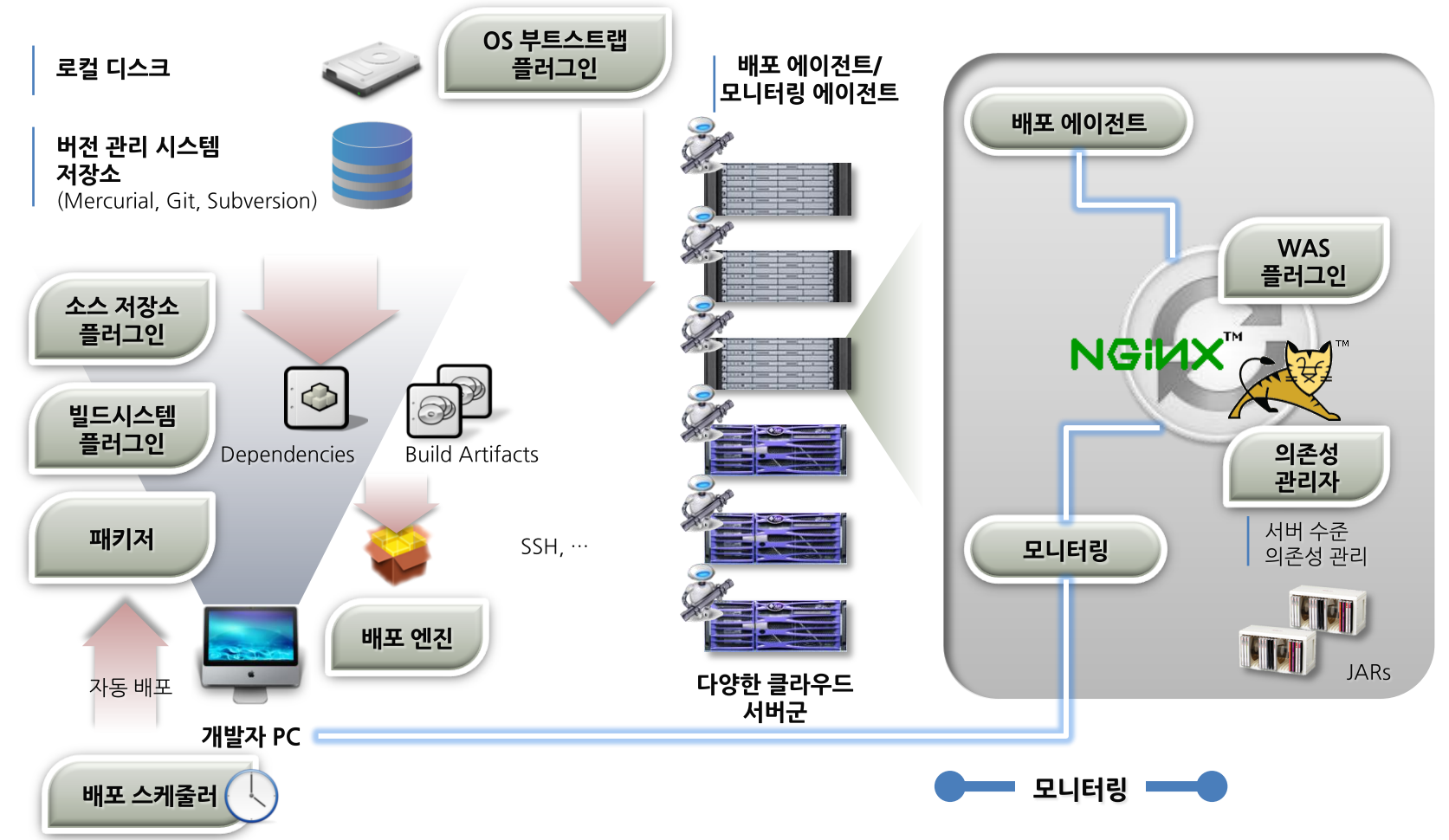
기능

- 소프트웨어 배포/설치
- 활성화/비활성화
- 업데이트
- 버전 추적
- 라이브러리 의존성 추적과 배포
- 일정 관리
- 모니터링

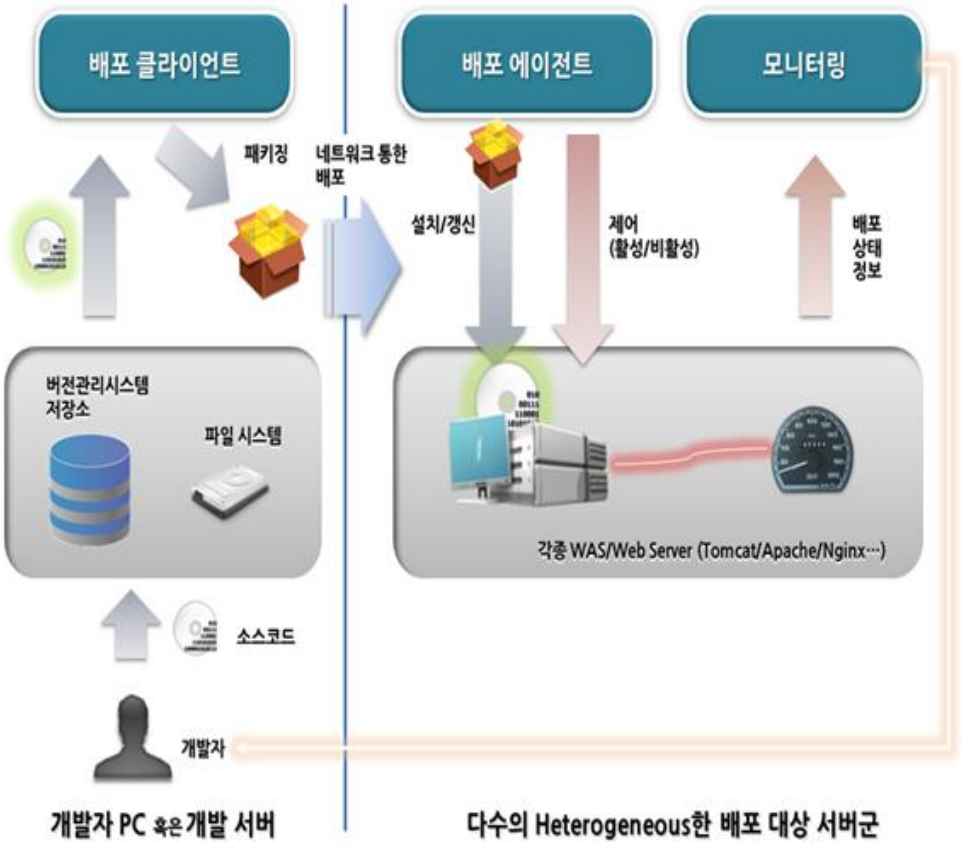
비기능

- **오픈소스 라이선스:** GPLv3
- **스택 지원:**
  Apache HTTPD, Nginx,
  Apache Tomcat
- **라이브러리/프레임워크 지원:**
  JVM 기반 언어/전자정부 프레임워크(Spring)
- **보안**
- **소프트웨어 배포 에이전트 부트스트래핑**

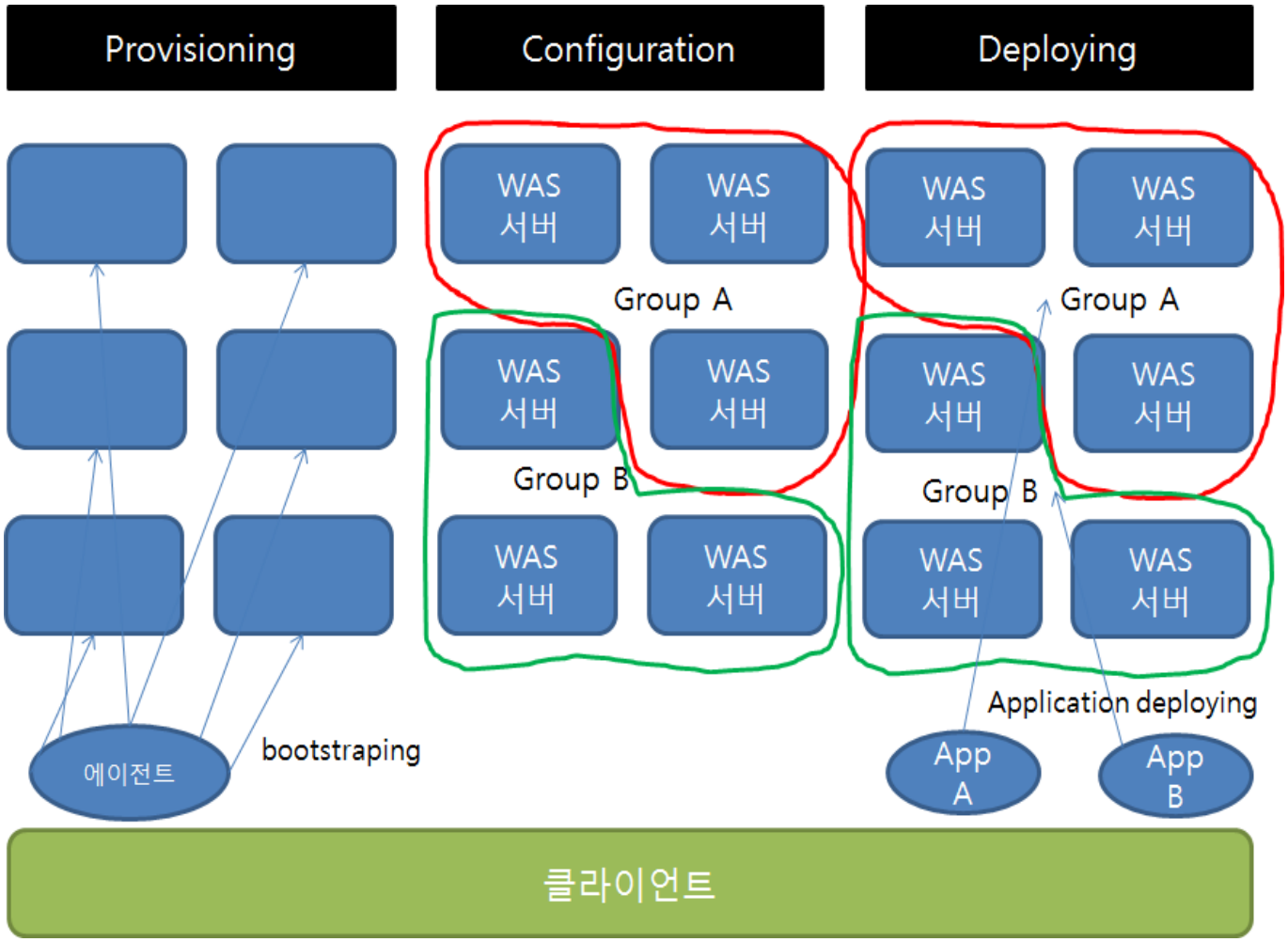
시스템 구조


배포 클라이언트
배포 에이전트
로컬 디스크
OS 부트스트랩 플러그인
배포 에이전트/ 모니터링 에이전트
버전 관리 시스템 저장소 (Mercurial, Git, Subversion)
배포 에이전트
WAS 플러그인
NGINX™
소스 저장소 플러그인
빌드시스템 플러그인
Dependencies
Build Artifacts
의존성 관리자
패키저
SSH, …
모니터링
서버 수준 의존성 관리
배포 엔진
JARs
자동 배포
다양한 클라우드 서버군
개발자 PC
배포 스케줄러
모니터링

아키텍처(1)


배포 클라이언트
배포 에이전트
모니터링
패키징
네트워크 통한 배포
설치/갱신
제어 (활성/비활성)
배포 상태 정보
버전관리시스템 저장소
파일 시스템
각종 WAS/Web Server (Tomcat/Apache/Nginx…)
소스코드
개발자
개발자 PC 혹은 개발 서버
다수의 Heterogeneous한 배포 대상 서버군

아키텍처(2)


Provisioning
Configuration
Deploying
WAS 서버
WAS 서버
WAS 서버
WAS 서버
Group A
Group A
WAS 서버
WAS 서버
WAS 서버
WAS 서버
Group B
Group B
WAS 서버
WAS 서버
WAS 서버
WAS 서버
Application deploying
에이전트
bootstraping
App A
App B
클라이언트

개발 내용

배포 에이전트

- 다양한 클라우드 지원

배포 엔진 개발
소프트웨어 배포, 설치, 제거, 활성/비활성화/업데이트와 관련된 명령 해석과 비즈니스 논리 처리

라이브러리 의존성 추적 및 소프트웨어 스택 구성 관리
사용하는 프로그램의 라이브러리 및 스택 구성 분석 및 설정

배포 클라이언트

- 멀티 플랫폼 지원
- 배포 에이전트 부트스트래핑

클라이언트 엔진 개발
(플랫폼 중립적) 배포 에이전트와 통신해 명령을 주고 결과를 받는 비즈니스 논리 처리

소스 코드 버전 관리 시스템 연동

파일 업로드 서비스

일정 관리

모니터링 도구

- 다양한 스택 연동

모니터링 엔진 개발
서버/배포 에이전트/스택 상태 확인, 소프트웨어 배포/설치 이력, 소프트웨어 활성/비활성 상태 등 외부 환경에 대한 감시를 수행할 비즈니스 논리 처리

보안 강화(AS-IS vs TO-BE)

에이전트(서버) 단	• whitelist 기반으로 접속 가능한 클라이언트 통제 • IP와 요청 명령을 기록하는 접속 로그 유지
클라이언트 단	• 인증서 기반으로 보안 채널을 통해 서버 접속(SFTP, SSH)



향후 계획	• 상용화 과정에서 보안 강화 요청이 들어올 경우 프로그램 업로드와 WAS 컨트롤을 이중 채널로 독립적으로 수행하도록 분리(특허)

과제 추진 실적

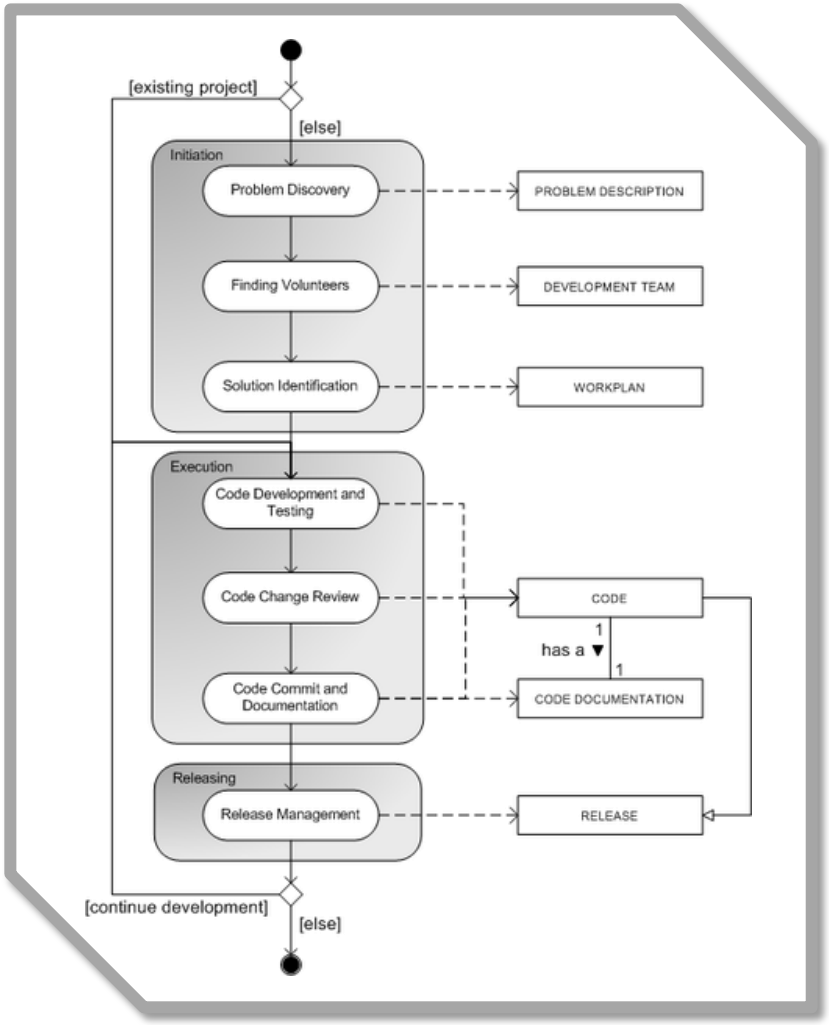
구현	배포 클라이언트와 배포 에이전트
오픈 소스 라이선스 검증	정보통신산업진흥원 공개SW 역량 프라자의 공개SW 라이선스 검증 서비스 완료(라이선스 모델: GPLv3)
소프트웨어 품질/기능 검증	내부 검증은 물론이고 STA테스팅 컨설팅과 함께 외부 품질/기능 검증까지 완료
패키징과 오픈소스 공개	구글 코드 저장소(https://code.google.com/p/ideployer/)에 ideployer라는 이름으로 루비 패키지(gem) 형태의 바이너리와 DVCS인 Hg로 다운로드 가능한 소스 코드 전체 공개:
특허 출원과 마케팅/상용화	특허 출원: 서비스 보안 인증 방법 및 이를 구현한 웹 애플리케이션 서버(10-2012-0142661) 마케팅: 도메인 이름 확보(ideployer.com). 회사 홈 페이지 리뉴얼 중

개발 방법론(1)

오픈 소스 개발 방법론에 따라
전체 개발 프로세스 정립

- 소스 코드 이력 관리, 위키, 이슈 관리
- 코드 검토: 중요 핵심 모듈 대상
- 언어: 코드/주석과 위키/문서는 모두 영어

open source



개발 방법론(2)



Ruby
A Programmer's Best Friend



Sinatra







Atlassian
bitbucket

Google™
code

상용화 및 발전 계획

오픈소스 라이선스 핵심 배포 시스템

저변 확대/개발자 참여 도출

상용화

관리 도구 고도화
IDE 통합
압축 통한 성능 개선

웹/VPS 호스팅, 퍼블릭 클라우드 환경에 최적화

고도화

PaaS 시스템으로 발전

프로그래밍 언어 지원 확대

스택 관리/사용자 관리 등 요구 사항 고려

서비스 관점에서 PaaS를 활용할 수 있는 상용 모델을 위해 가상화 환경에 중립적인 방향으로 개발

상품으로서 부가가치 증대

- 전자정부 표준 프레임워크와 연계
- JSP나 자바 호스팅 환경에 부가가치 제고
- AWS API를 적극적 활용

시장 진출 방안

- 국내 클라우드 서비스 업체에 맞는 최적화 작업
  → 업체와 연계해 일반 고객에 서비스
- 국내 대형 기업 대상 맞춤 솔루션 제공
- 해외 클라우드 업체에 맞는 솔루션 상용,
  ISV 자격 획득 후 해외 시장 개척

상용화를 위한 준비 사항

내부 프로젝트 활용

이클립스/스프링/메이븐 기반 금융권 프로젝트에 활용 중

외부 협력 강화

다양한 사이트에 배포가 필요한 자바/스프링 기반 솔루션 업체 한 곳에 프리 세일즈



상용화 준비

내부 활용 경험과 외부 피드백을 바탕으로 상품화 진행

- 사용자 인터페이스 강화
- 각종 오픈 소스 도구(예: Chef)와 결합

기대 성과

기술 측면

분산 컴퓨팅 환경 지원 배포 기술 개발

- 수 많은 서버 대상 배포 관리 기술 개발
- 배포 과정 자동화 및 모니터링 소프트웨어 개발

클라우드 환경 기반 배포 기술 개발

- 퍼블릿, 프라이빗, 하이브리드 환경을 모두 아우르는 클라우드향 배포 기술 개발
- PaaS 솔루션의 기본 모듈로 응용 가능

경제 산업 측면

서비스 리드타임 최소화로 경쟁력 강화

- 제품 개발 주기를 짧게 가져갈 수 있음
- 개발 배포에 들어가는 비용 절감 및 매출 증대 기대

개발 생산성 향상

- 개발 및 테스트 시간 단축
- 개발 운영 비용 절감

감사합니다.

㈜이노디에스

www.innods.com

박재호 이사

E. jaypark@innods.com

T. 02-6205-1320