

# **Open Source Plattform: Enterprise Information Logistics Framework**

Eclipse Project Proposal

# Überblick

- Hintergründe
- Entscheidung für Eclipse & CSD
- Anforderungen
- Framework-Architektur
- Konzernumgebungen
- Anwendungsszenarien

# Hintergründe

- Management von unstrukturierten Daten
- Nichtexistenz vom Standard Framework
- Probleme mit proprietären Lösungen

## Daten in Unternehmen



- 6 Terabyte an strukturierten Daten
- 24 Terabyte an **unternehmens-kritischen** unstrukturierten Daten

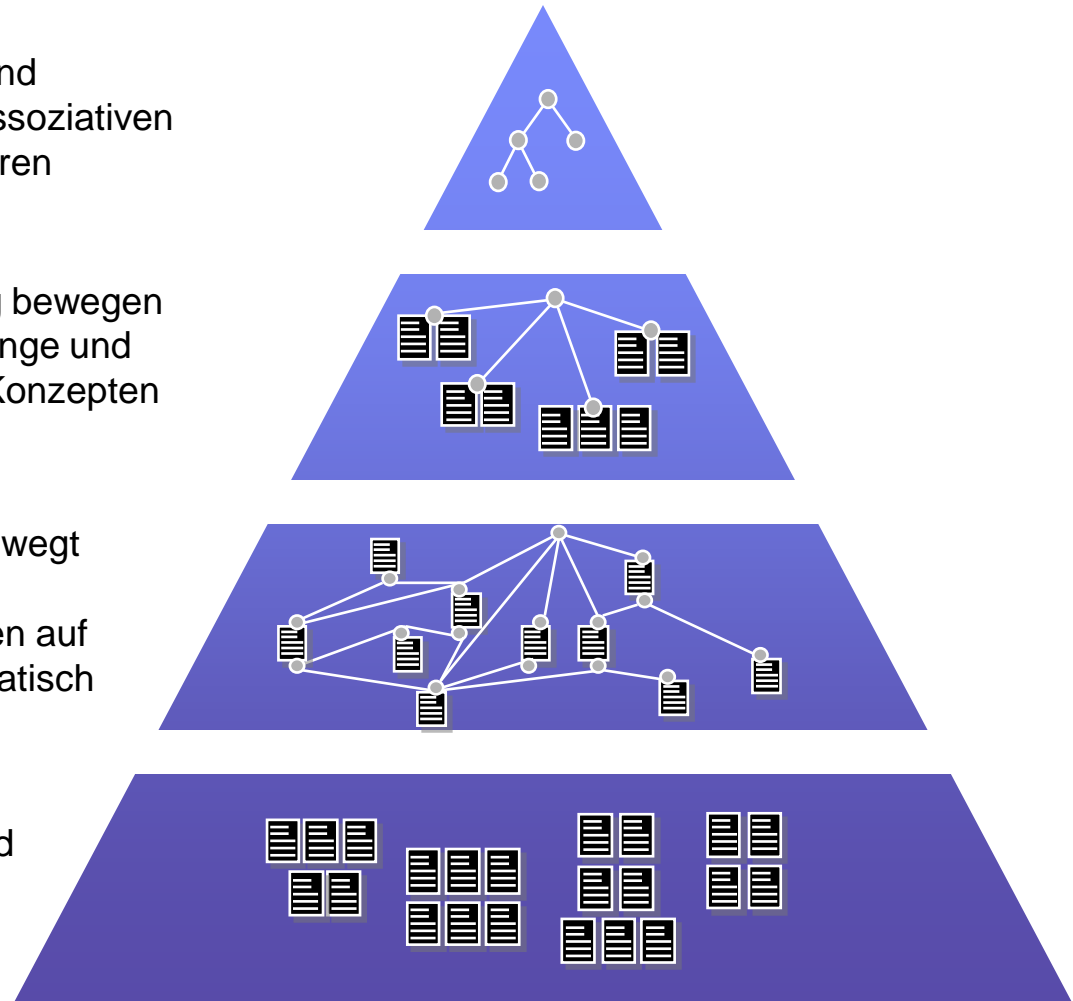
# Suchverfahren

**Entscheidungsbäume** bilden Prozesse und Workflows ab, die auf strukturierten und assoziativen Such- und Klassifizierungsverfahren basieren

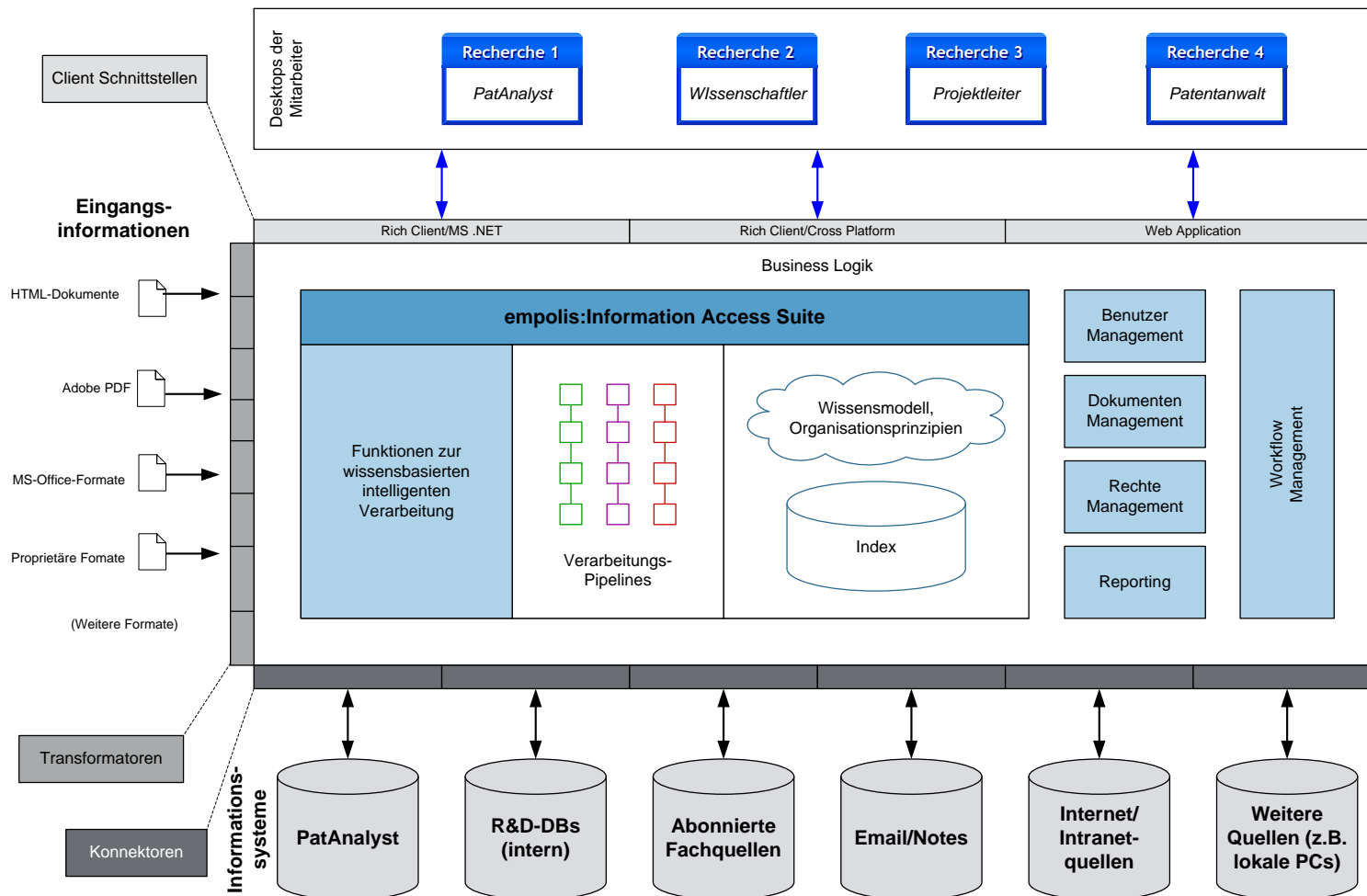
**Strukturierte Suche und Klassifizierung** bewegen sich auf der Wissens Ebene. Zusammenhänge und Ähnlichkeiten werden auf der Ebene von Konzepten modelliert und ausgenutzt.

**Assoziative Suche** und Klassifizierung bewegt sich auf der Informationsebene. Zusammenhänge und Ähnlichkeiten werden auf Ebene von Begriffen/Wörtern rein mathematisch erkannt und hergestellt.

**Volltextsuche** bewegt sich auf der Datenebene. Durch formale Ausdrücke und Anfragen werden Vorhandensein und Übereinstimmungen von Ziffernfolgen ausgewertet.



# e:IAS Architekturüberblick



# Hintergründe

- Proprietäre Lösungen
  - Schwer zu implementieren, warten und erweitern
  - Ständig das Rad neu erfinden
  - Langsame Innovation
  - Lange Entwicklungszeiten
  - Standard-Konformität?
  - Flexibilität?

# Entscheidung für Eclipse & CSD

- Mission
  - Etablierung vom Standard und technologischer Infrastruktur für die nächste Generation von Informationsmanagementsystemen
- Ziel
  - Konzeption und Implementierung von einem Standard Framework im Bereich vom Informationsmanagement
- Aufbau der Community für das Einbringen von verwandten Aspekten wie
  - OCR
  - Data-Mining
  - Übersetzen von Dokumenten
  - Management vom Rich Media

# Entscheidung für Eclipse & CSD

- Eclipse
  - Sehr große und aktive Community
  - Geschäftsfreundliche Lizenz
  - Sicherheit durch IP-Prozess
  - Klare und kontrollierte Prozesse
  - Qualität und Integrität von Projekten
  - Marke

# Entscheidung für Eclipse & CSD

- empolis GmbH & brox IT-Solutions GmbH
  - > 25 Jahre Erfahrung
  - > 50 Experten
  - eccenca Foundation e.G.
  - Fokussierung auf Kernkompetenzen
  - Professionelles Support und Serviceangebot

# Anforderungen

- Funktionelle Anforderungen
- Nicht funktionelle Anforderungen

# Funktionelle Anforderungen (1/2)

- Benutzung von Standards
- Komponentisierung
- Zentrales Management
- Security
- Deployment-Flexibilität
- Erhaltung von verarbeitenden Informationen

## Funktionelle Anforderungen (2/2)

- Implementierungsspracheneutralität
- Inkrementeller Indexupdate
- Die Suche verändert den Status nicht
- Begrenzte bidirektionale Kommunikation zwischen Komponenten
- Buffering vom externen Informationsfluss
- ...

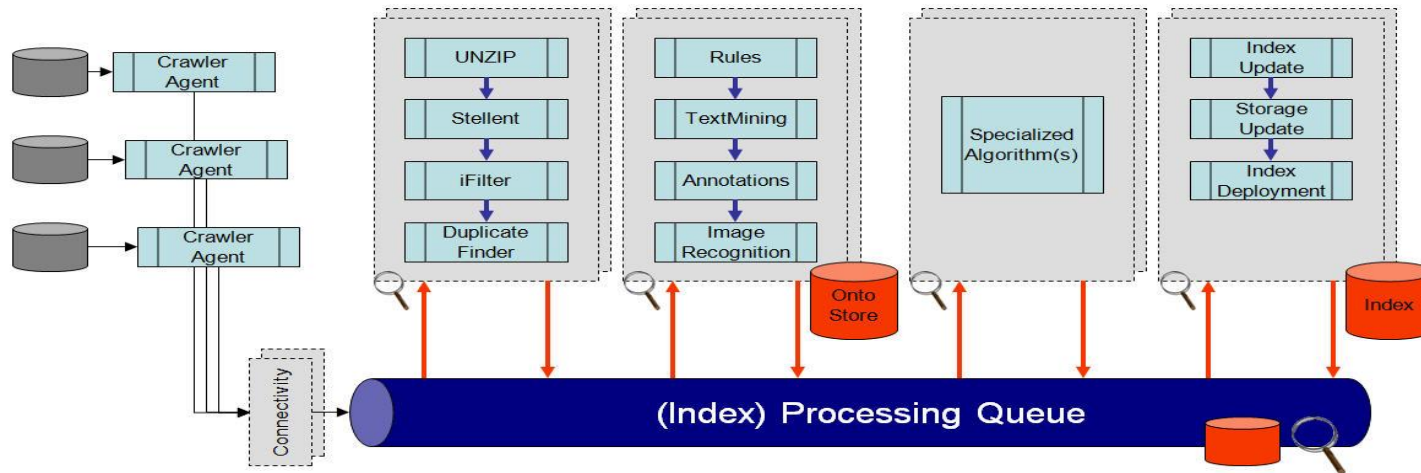
## Nicht funktionelle Anforderungen (1/2)

- Deployment auf preiswerter Hardware
- Skalierbarkeit
- Ausfallsicherheit
- Robustheit
- Datenkonsistenz

## Nicht funktionelle Anforderungen (2/2)

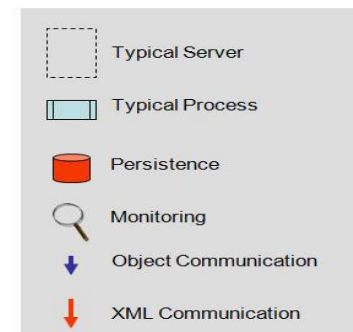
- Live Update von Komponenten
- Copy-Deployment
- Hoher Datendurchsatz beim Indizieren
- Community und Partner freundlich
- Bedienbarkeit
- ...

# Architekturübersicht



## Key Ideas

- Crawlers/Agents push data into Connectivity / Entry Point
- Connectivity Module filters, converts versions, extracts binaries etc. and pushes into queue
- Message-driven queue stores data and guarantees delivery
- 1...n servers respond to messages, process data and write back to queue
- Potentially multiple instances of servers for load balancing and increased throughput
  - Open issue: synchronization of persistence
- 1...n processes inside server arranged via BPEL (~pipelines, ~strategies)
- Search yet to be defined separately but the objective is to separate the processes of (a) filling the index and (b) using the index for search (unlike in e:IAS)



# Basistechnologien

- OSGi/SCA als Komponentenmodelle
- Message Queue
- BPEL
- XML
- Storage (XML, verteiltes Dateisystem, ...)
- Suchtechnologien sowie KM Technologien
  - Beispiel: Lucene, IBM, Fast, Google, ...
  - Extraktionstechnologien
    - GATE
    - Dokumentenkonverter (z. B. Apache POI, Stellant, ...)
  - Extreme Diversifikation von Technologieanbietern ( > 2000)

# Konzern (Anwendungen und Betrieb)

- Anwendungen
  - Verantwortlichkeiten (z. B. Betrieb, Entwicklung, ...)
  - Besitzer der Applikationen (Abteilung / Fachbereich)
  - Optimale Funktionalität
  - Implementierung
  - Know-how bezüglich der Technologien
  - Nachhaltigkeit (Standardisierungsabteilungen...)
  
- Betrieb
  - Einfache Wartung
  - Lernkurve

# Konzern (Status quo)

- Verschiedene Technologien (z. B. > 70 Suchtechnologien bei einem größeren Konzern)
- Kosten
  - Implementierungskosten
  - Bis zu 5 mal höhere Wartungskosten
- Investitionssicherheit
  - Standarisierungsabteilungen
- Eingeschränkte Kommunikation
  - Zonenmodelle
  - Firewalls
  - Protokolle

# Konzern (Technologieeinführung)

- Konzerndurchdringung von z. B. Suchtechnologie
  - 5 % geschafft → Neue Strategie
  - Sind Investitionen dann verloren?
- Lernkurven
  - Wie werden Mitarbeiter ausgebildet?
  - Kann Wissen übertragen werden?
- Wie kann das Problem entschärft werden?
  - Standardisierung → Framework
    - Technologiehersteller
    - Konzern
    - Verstehen wir uns in einer „Pre-JDBC/ODBC“ Ära!

# Anwendungsszenarien

- Einfache Anwendungen (Building Block)
- Search Appliance
- Search Building Block auf Infrastruktur Ebene
- Abdeckung der Anwendungsszenarien durch **ein** Framework

# Anwendungsszenario

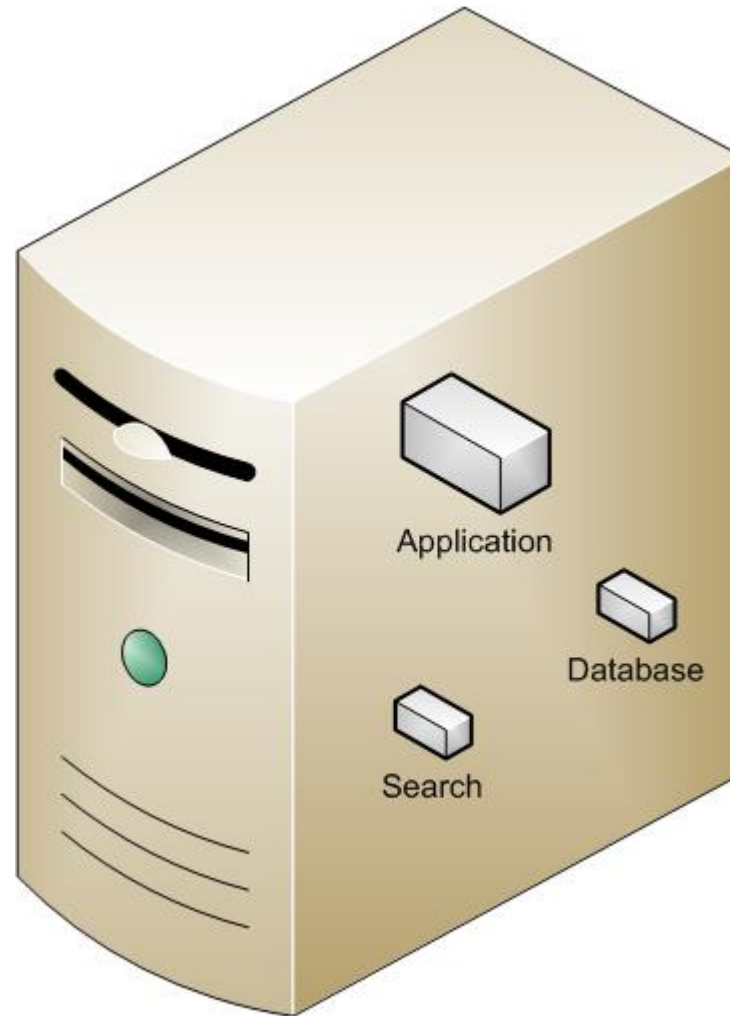
## - Einfache Anwendung

Funktionen:

- Einfache Anwendung
- Suche als Baustein auf dem Rechner installiert
- Bequemer und einfacher Einstieg
- Migrationsstrategien auf erweiterte Szenarien werden geschaffen
- Suche kann mit der Anwendung wachsen
- Integration einfach und effektiv
- Schaffung hochwertiger Daten

Einsatz:

- Suche
- Dublettenbereinigung/-vermeidung
- Database Offload
- ...



Application Building Block - Search

# Anwendungsszenario

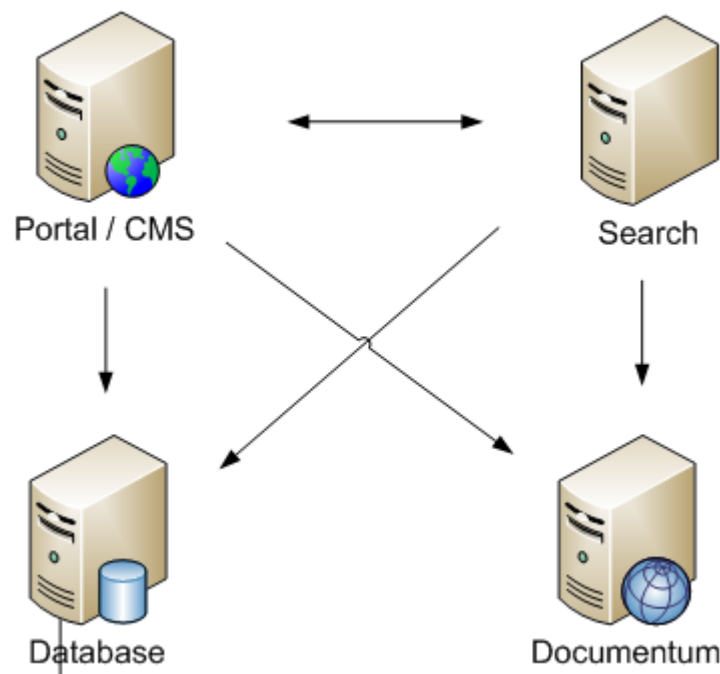
## Search Appliance

Funktionen:

- Einfache Anwendung
- Einfache Installation der Suche auf einem Rechnercluster
- Einfache Migration zum Infrastruktur Building Block
- Suche kann mit der Anwendung wachsen
- Bequemes und effektives Management
- Integration einfach und effektiv
- Schaffung hochwertiger Daten

Einsatz:

- Suche
- Erzeugung von Metadaten
- Basis für Services
- ...



# Anwendungsszenario

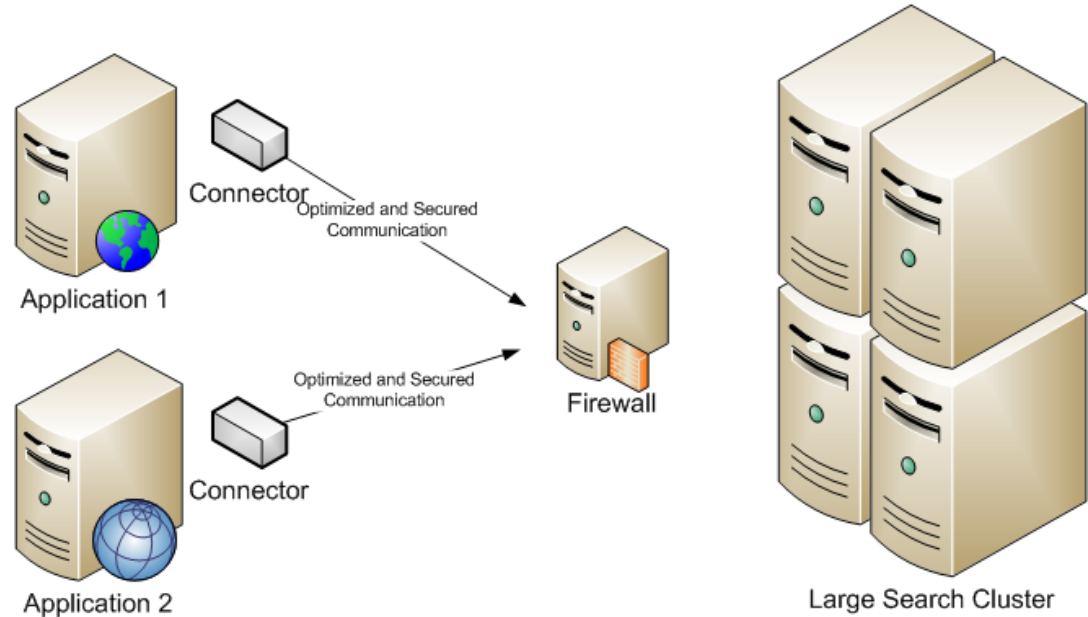
## Infrastruktur Building Block Suche

Funktionen:

- Zentraler KM Block innerhalb eines Konzerns
- Bequemes und effektives Management
- Schaffung hochwertiger Daten
- Zentrales Aufwerten von Daten

Einsatz:

- Suche
- Erzeugung von Metadaten
- Basis für Services
  - Business Workflows
  - Entscheidungsunterstützung
  - ...
- ...



# Fazit: Ein Framework

- Kunden
  - Standardisierung der Infrastruktur
  - Mehr Funktionen durch Community
  
- Technologieanbieter
  - Verlust eines extremen Kostentreibers
  - Mehr Funktionen durch Community
  
- Forschung
  - Schnellere Innovationszyklen

# Unsere Aktivitäten

- „Creation Review“ bei Eclipse
- Projekt Übersicht
  - aktuell 12 Entwickler
  - Konzepte
  - Erste Prototypenimplementierung
- Vorbereiten einer ersten downloadfähigen Version

# Projekt

- Kontakt:
  - August Georg Schmidt, brox IT-Solutions GmbH
  - Igor Novakovic, empolis GmbH
  
- Ressourcen
  - <http://www.eclipse.org/proposals/eilf/>
  - Newsgroup: eclipse.technology.eilf

Vielen Dank!