



**Where we come from  
Where we are  
Where we're going**

DD4T Roadmap

# Where we've come from

---

- Started in September 2011
- Launching customer SKF
- Java, followed by .NET
- Discussion group on Google Groups:  
<http://groups.google.com/group/dynamic-delivery-4-tridion>
- First code check in (SVN): September 20, 2011
- First developers: Rogier Oudshoorn, Bjorn Lindstrom, Quirijn Slings

# Where we are

---

- 35+ customers in 7 countries
  - 1/4 on Java
  - 3/4 on .NET
  - 1/3 in the US
  - 2/3 in EMEA
- Version 1.31
- Community: 37 committers, 6 active in the last 12 months
- 953 changes in last 12 months (100 commits)
- Google discussion group:
  - <http://groups.google.com/group/dynamic-delivery-4-tridion>
- Recently moved from Google Code to GitHub
  - <https://github.com/dd4t/dynamic-delivery-4-tridion>

# DD4T Mission

---

- Open source framework
- Supports .NET and Java
- Provide access to Tridion items as POJOs or POCOs
- Link to the dominant MVC architecture
  - Currently: ASP.NET MVC 3-5, Spring MVC 3
- Stay as small as possible

# Features: .NET

---

- Generic content model
- Factory / provider pattern
- Handles links and rich text
- Serving binaries from Content Broker database
- Labels (resx)
- Sitemap (navigation)
- Experience manager support
- Supports ASP.NET MVC 3, 4 and 5
- Supports Tridion 2009 SP1, 2011 GA, 2011 SP1, 2013 GA and 2013 SP1

# Features: Java

---

- Generic content model
- Factory / provider pattern
- Handles links and rich text
- Integration with Spring MVC 3
- Supports Tridion 2009 SP1, 2011 GA, 2011 SP1, 2013 GA and 2013 SP1

Example site:

- Serving binaries from Content Broker database
- Experience Manager (XPM) support

Where we're going

---

**DD4T**

**2.0**

# Planning

---

- Q4 2014: release 2.0 for java
- Q1 2015: release 2.0 for .NET
  
- General goals:
  - Improve maturity
  - Improve ease of use for non-Tridion developers
  - Enhance performance
  - Support as many SDL products/add-ons as we can



# New features in DD4T Java 2.0

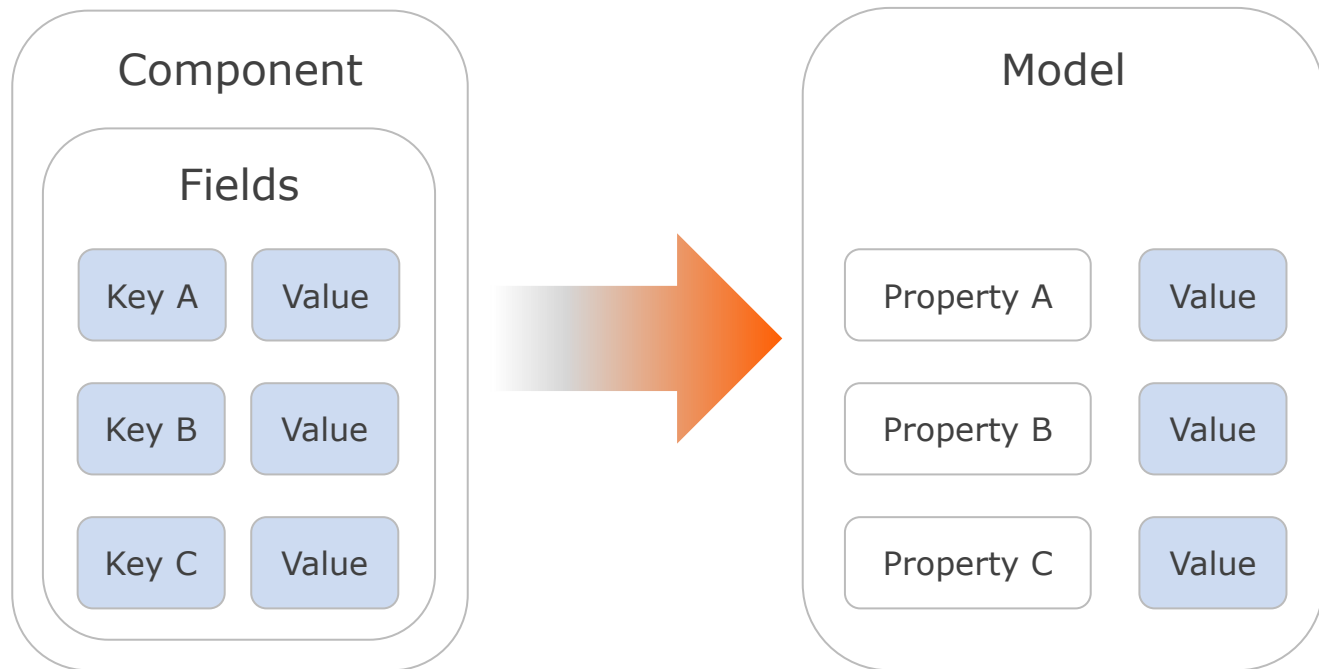
---

**Feature set is still being discussed, but will probably include most of the following:**

- Experience Manager support and binary handling in framework
- Cleaner separation of factories and providers
- Strongly typed models
- Caching using JMS-based invalidation
- Add basic controller implementations (currently only in example site)
- Move towards 'component presentation' as main model for retrieving dynamic content
- Support for REST-based access to the broker database
- Support for different data formats, including compressed JSON
- SmartTarget support

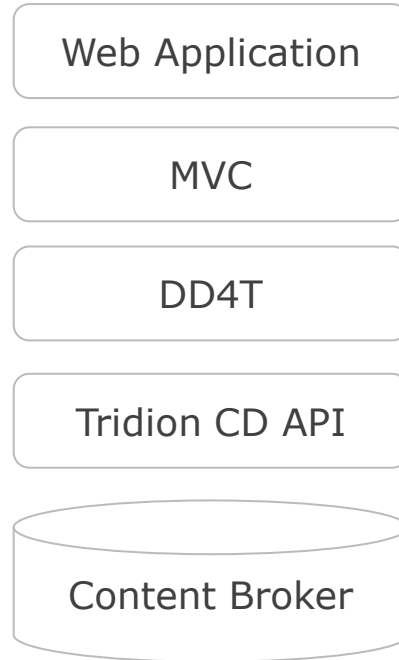
# Feature: strongly typed models

---



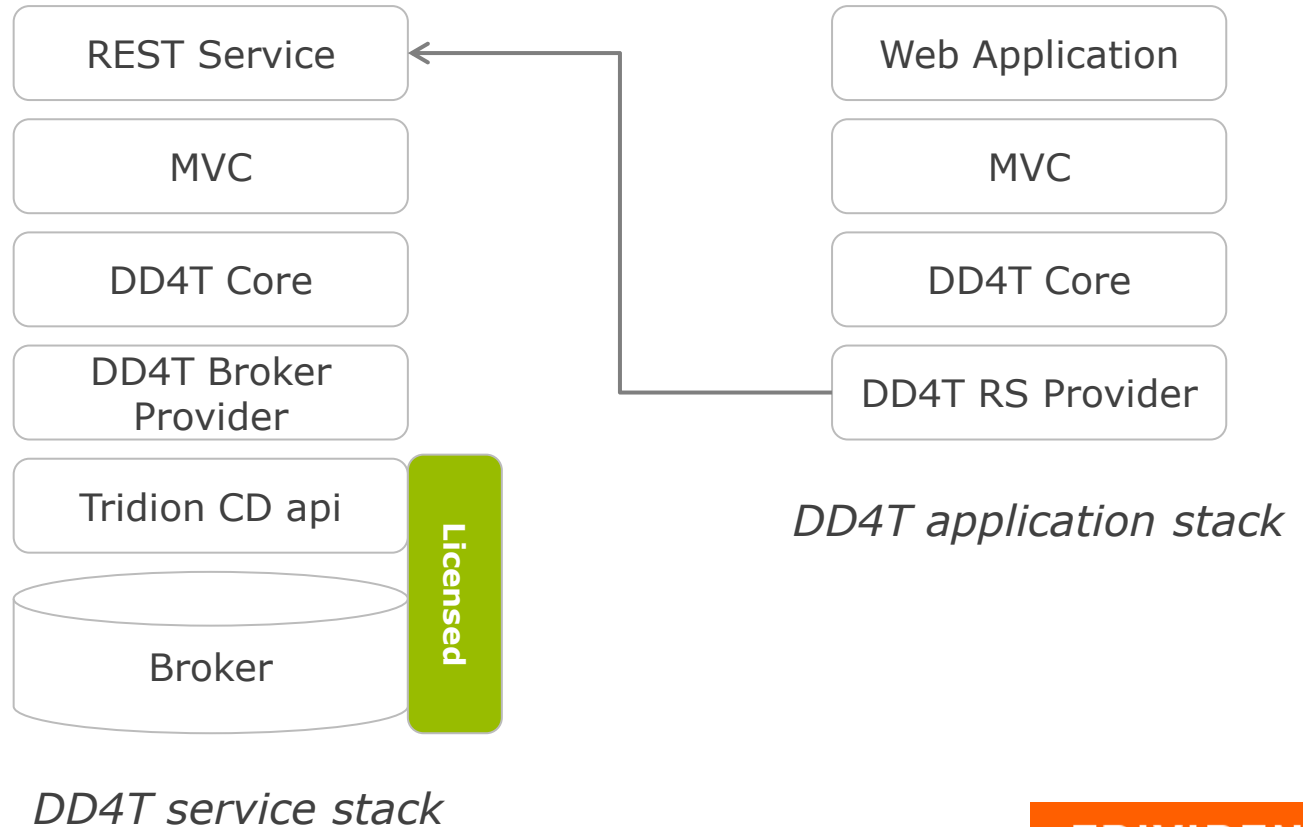
# Feature: REST service

---



*Typical DD4T Stack*

# Feature: REST service



# New features in DD4T .NET 2.0

---

## **Still to be discussed, but may well include:**

- Strongly typed models
- Caching using JMS-based invalidation
- Add basic controller implementations (currently only in example site)
- Move towards 'component presentation' as main model for retrieving dynamic content
- Support for REST-based access to the broker database
- Support for different data formats, including compressed JSON
- SmartTarget support