



Enterprise UI Architecture Framework

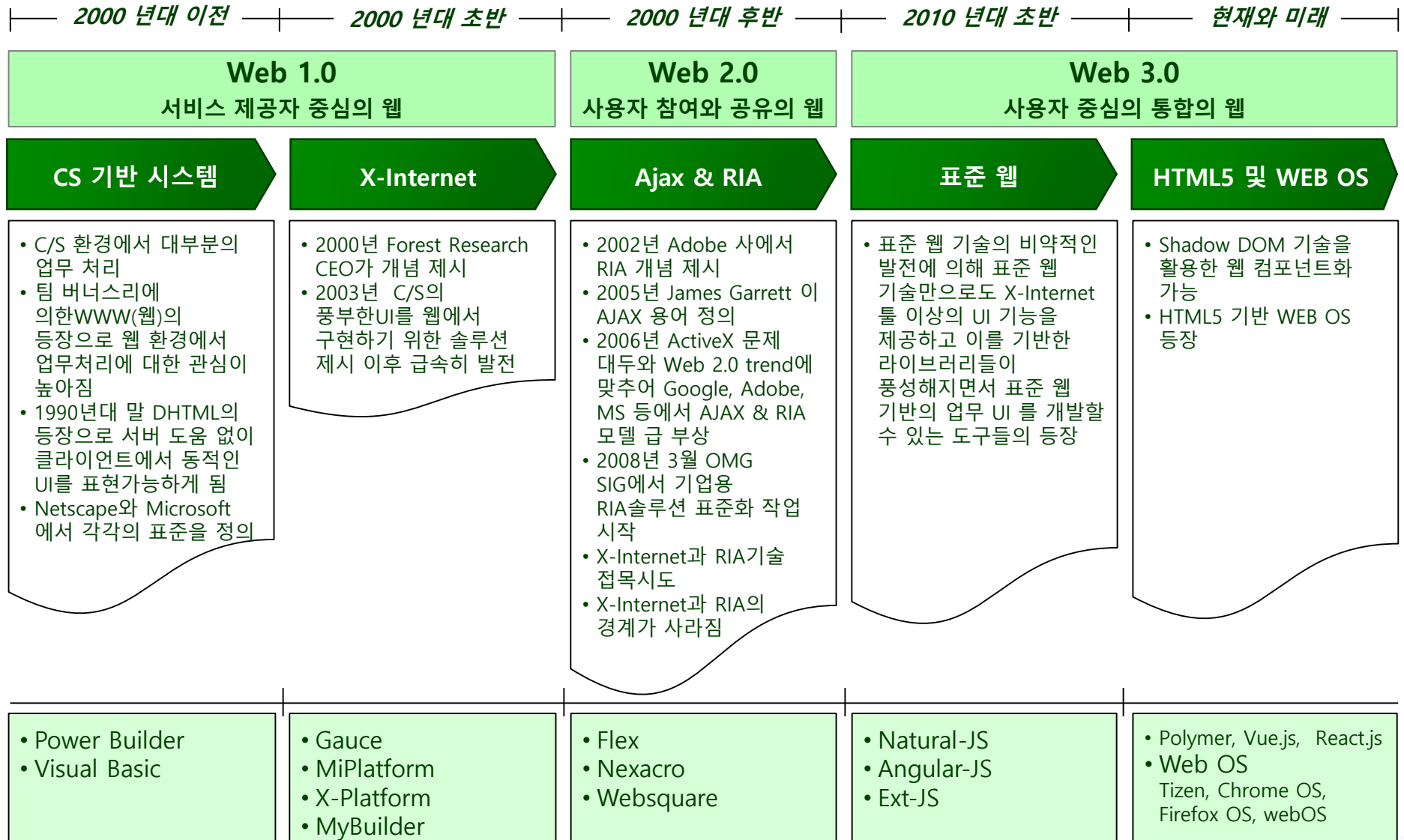
Natural-JS

2018.10.

목 차

1. 엔터프라이즈 어플리케이션 UI 개발 기술의 변화
2. 현 표준 웹 기반 엔터프라이즈 UI 개발도구 현황
3. Natural-JS 아키텍처 구성
4. Natural-JS 특징
5. 기대효과

1. 엔터프라이즈 어플리케이션 UI 개발 기술의 변화

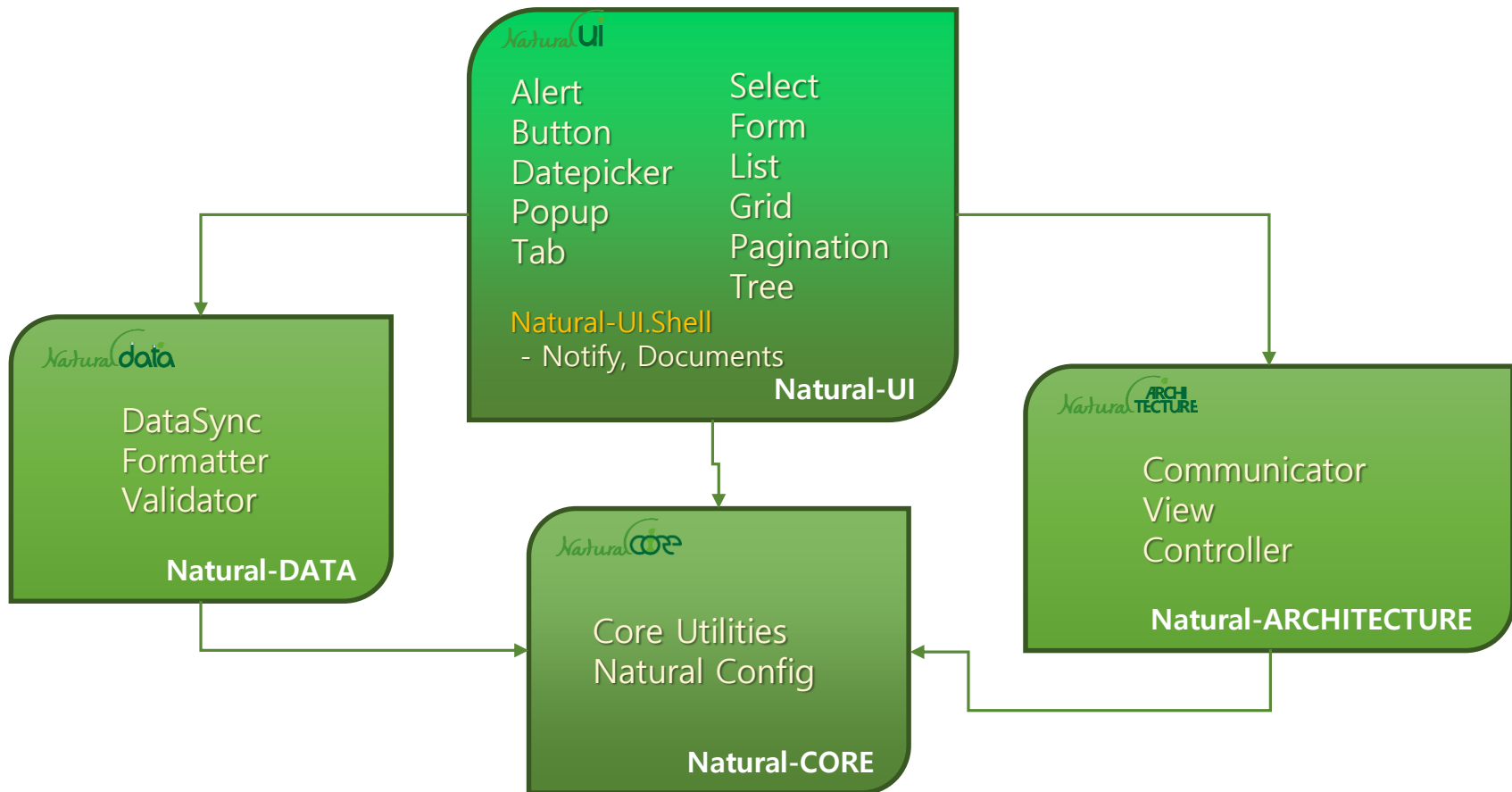


2. 현 표준 웹 기반 엔터프라이즈 UI 개발도구 현황

구분	Natural-JS	AngularJS (Google)	Vue.js	React.js (Facebook)	SAPUI5 (SAP)
아키텍처	<ul style="list-style-type: none"> CVC 패턴기반 AOP 지원 페이지 블록 단위 프로그래밍 지원 	<ul style="list-style-type: none"> MVC 패턴 기반 DI 지원 	<ul style="list-style-type: none"> MVVM 패턴 기반 Virtual DOM 	<ul style="list-style-type: none"> Flux 패턴기반 Virtual DOM 	<ul style="list-style-type: none"> MVC 패턴기반
데이터	<ul style="list-style-type: none"> 컴포넌트간 양방향 데이터 바인딩 지원 강력한 포맷, 마스킹, 검증 모듈 제공 	<ul style="list-style-type: none"> 양방향 데이터 바인딩 지원 	<ul style="list-style-type: none"> 단방향 데이터 바인딩 지원 	<ul style="list-style-type: none"> 단방향 데이터 바인딩 지원 	<ul style="list-style-type: none"> 양방향 데이터 바인딩 지원 포맷, 검증 지원
UI 컴포넌트 지원 점수 (10)	<ul style="list-style-type: none"> 9 	<ul style="list-style-type: none"> 9(서드파티를 통해 지원) 	<ul style="list-style-type: none"> 9(서드파티를 통해 지원) 	<ul style="list-style-type: none"> 9(서드파티를 통해 지원) 	<ul style="list-style-type: none"> 10
디자인, 퍼블리셔와의 협업	<ul style="list-style-type: none"> 매우 용이함 	<ul style="list-style-type: none"> 어려움 	<ul style="list-style-type: none"> 어려움 	<ul style="list-style-type: none"> 어려움 	<ul style="list-style-type: none"> 매우 어려움
학습곡선	<ul style="list-style-type: none"> 심플한 UI 아키텍처와 jQuery 플러그인 형태로 기능들을 제공하여 학습 곡선이 낮음. 	<ul style="list-style-type: none"> 아키텍처 스펙을 이해 해야 하고 생소한 구문들이 많아 학습 곡선이 높음 	<ul style="list-style-type: none"> 아키텍처 스펙을 이해 해야 하고 생소한 구문들이 많아 학습 곡선이 높음 	<ul style="list-style-type: none"> 아키텍처 스펙을 이해 해야 하고 생소한 구문들이 많아 학습 곡선이 높음 	<ul style="list-style-type: none"> SAP 아키텍처를 이해하는 웹 개발자에게는 학습 곡선이 낮음.
구동환경	<ul style="list-style-type: none"> ECMA Script 3+(Suitable 5+) HTML4+ jQuery 	<ul style="list-style-type: none"> ECMA Script 3+(Suitable 5+) HTML4+ jQuery Lite 	<ul style="list-style-type: none"> ECMA Script 5+ HTML5 	<ul style="list-style-type: none"> ECMA Script 5+ HTML5 	<ul style="list-style-type: none"> ECMA Script 5+ HTML 4+

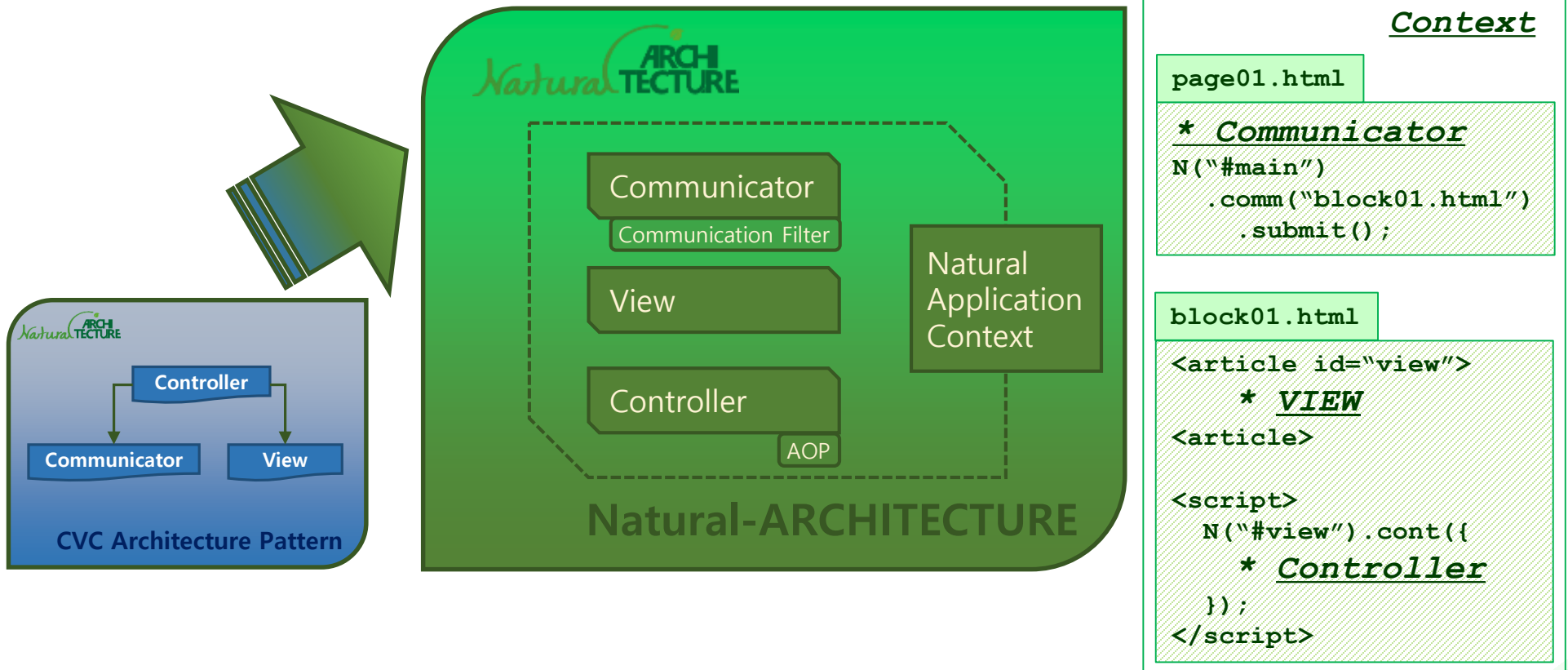
3. Natural-JS 아키텍처 구성

- Natural-JS 는 Natural-CORE, Natural-ARCHITECTURE, Natural-DATA, Natural-UI 로 구성 됩니다.



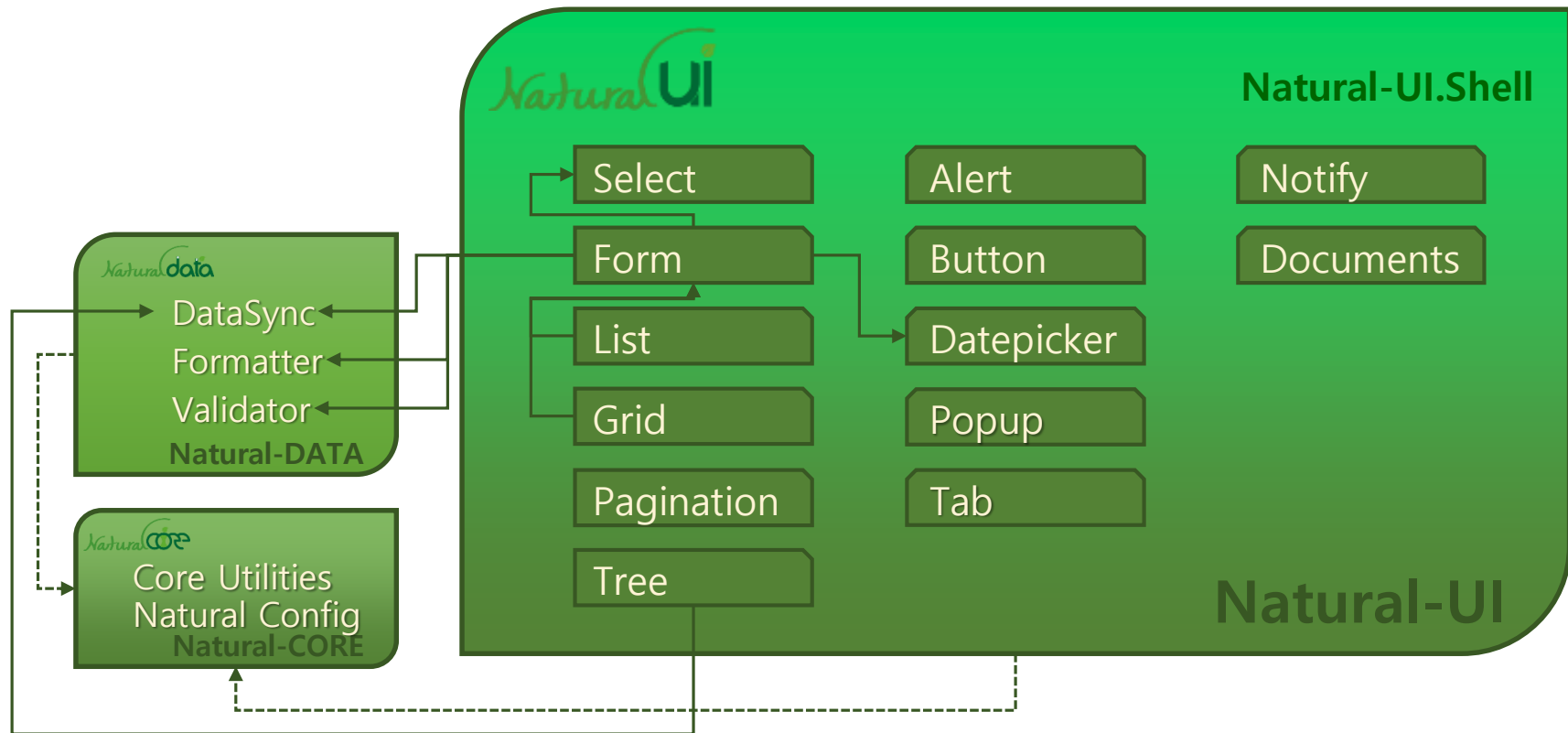
3. Natural-JS 아키텍처 구성

- **Natural-ARCHITECTURE** 는 **CVC(Controller-View-Communicator)** 아키텍처 패턴(김항만, 김용구, 한국통신학회, 9월 2011년) 을 구현한 아키텍처 프레임워크를 제공하여 웹 어플리케이션 UI 개발의 복잡성을 해결하고 개발 생산성 향상 시켜주며 개발된 UI 자원을 재 활용 할 수 있는 아키텍처를 제공 해 줍니다.



3. Natural-JS 아키텍처 구성

- **Natural-UI** 는 표준 웹 기반의 **Rich** 한 엔터프라이즈 웹 어플리케이션을 쉽게 구현 할 수 있도록 다양한 컴포넌트를 제공 하고 **Natural-DATA** 는 데이터의 검증(Validate) 및 양식화(Format), 데이터 동기화등 데이터 처리를 위한 기능을 제공 합니다. **Natural-CORE** 는 프레임워크 내부적으로 사용하는 공통함수와 웹 개발을 지원하는 함수들을 제공 합니다. 또한 이들은 상호 연동되어 작동 됩니다.



4. Natural-JS 특징

➤ 웹 UI 개발 생산성 향상

- **AOP(Aspect Oriented Programming)** 와 **Communication Filter** 기능을 제공 하여 함수 들이 실행 되는 특정 시점이나 서버로 데이터/페이지를 요청 할 때 마다 공통 로직들을 간단 하게 실행 시킬 수 있습니다.
- **데이터 관련 UI 컴포넌트들은 내부 데이터세트를 따로 관리**하며 여러 개의 컴포넌트들이 같은 하나의 데이터세트를 공유한다면 이들 **컴포넌트 간 데이터 값 및 표현이 자동으로 동기화** 되어 데이터 동기화 처리를 따로 하지 않아도 돼 데이터 동기화에 소비되는 개발 시간을 단축 시켜 줍니다.
- Formatter, Validator등의 비즈니스 어플리케이션 개발을 지원하는 모듈 들과 다양한 UI 컴포넌트들을 지원하여 개발 생산성을 향상 시켜 줍니다.

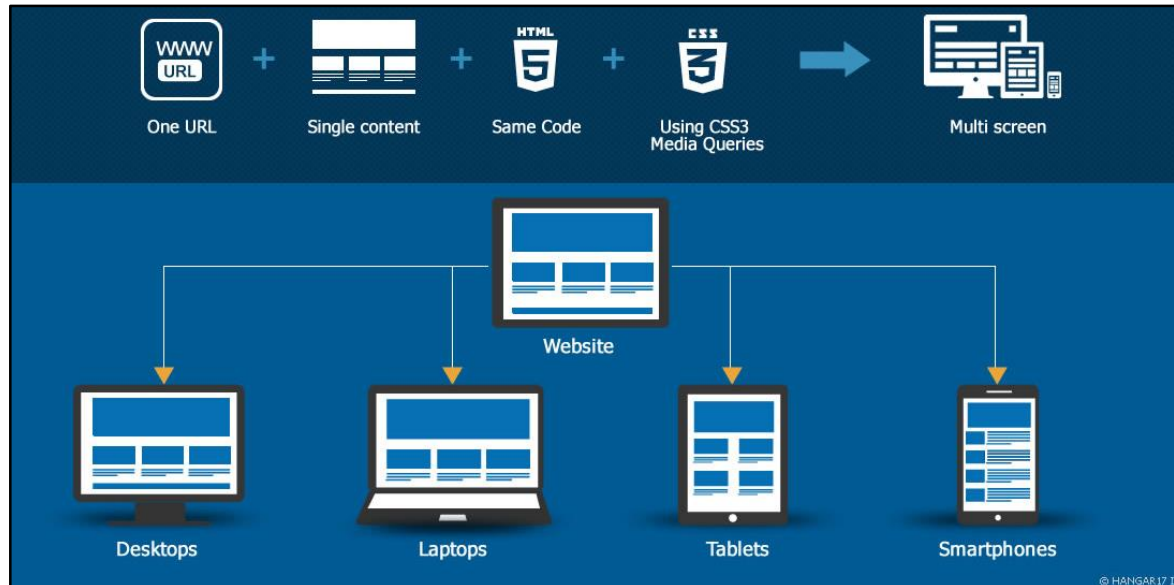
➤ 뛰어난 데이터 처리 성능과 반응 속도

- UI 컴포넌트들은 브라우저에서 제공하는 기본 기능들을 최대한 활용하여 UI 반응 속도가 빠릅니다.
- 프레임워크 라이브러리의 용량이 매우 작아(전체:208KB) 초기 로딩 속도가 매우 빠릅니다.
- Grid, List, Pagination 등의 컴포넌트를 제공하여 대용량 데이터를 빠르게 처리 할 수 있습니다.
- 여러 프로젝트의 성능점검을 통해 데이터 처리 및 UI 렌더링, 이벤트 바인딩 부분에 대한 튜닝이 여러 차례 이루어져 속도가 최적화 되어 있습니다.

4. Natural-JS 특징

➤ 웹 UI 안에서 개발영역과 디자인영역을 완벽하게 분리할 수 있는 아키텍처 제공

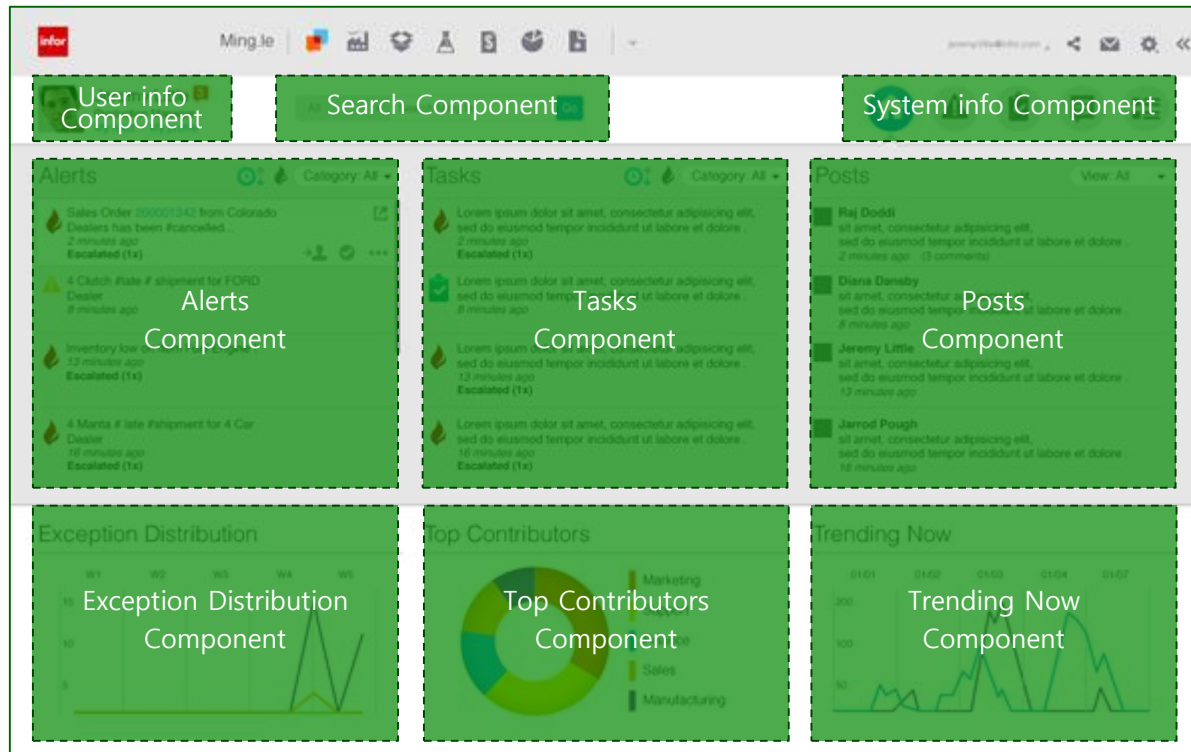
- Natural-JS 의 Natural-ARCHITECTURE 패키지는 클라이언트 브라우저 영역 안에서 프리젠테이션(디자인) 영역과 개발영역을 완벽하게 분리할 수 있는 프레임워크를 제공하여 디자인(퍼블리싱)은 디자인 파트에서 로직 개발은 업무별 UI 개발파트에서 따로따로 개발을 진행 할 수 있는 환경을 제공 해 줍니다.
- 표준기술 기반의 개발영역과 디자인영역을 분리 할 수 있는 아키텍처를 제공하여 이미 Natural-JS 로 개발 된 화면 들을 비즈니스 로직(스크립트) 수정 없이 HTML 과 CSS 만 수정 함으로서 반응형 웹, 모바일 웹 또는 모바일 하이브리드 앱 형태로 서비스 할 수 있습니다.



4. Natural-JS 특징

➤ 소스코드 재 활용

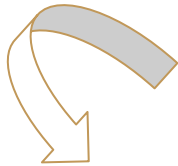
- Natural-JS의 Natural-ARCHITECTURE 패키지는 프로그램의 구동영역(Scope)을 완벽하게 분리 해 주어 SPA(Single Page Application)개발을 넘어 페이지 안의 페이지 블록들을 모두 웹 컴포넌트화 하여 재 활용할 수 있는 기반을 제공 합니다.



4. Natural-JS 특징

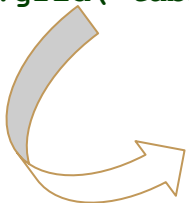
➤ 템플릿 기반의 UI 컴포넌트 아키텍처

- 퍼블리싱된 HTML 요소(템플릿 요소) 및 스타일을 기반으로 컴포넌트를 생성 해 줘 컴포넌트 스타일마저 별 다른 작업 없이 사이트의 전체적인 디자인 컨셉을 맞출 수 있습니다.
- 폼이나 그리드 컴포넌트는 템플릿 요소에 데이터 입력 컨트롤을 배치하고 포맷틀과 검증룰등을 선언 해 놓 으면 컴포넌트 실행 시 이러한 배치나 설정들이 내부적으로 인식되어 데이터의 입/출력 및 검증, 포맷등을 아주 쉽게 처리 할 수 있습니다.



<input type="checkbox"/>	eyeColor	name	company	gender
	index	email	registered	isActive
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

`N(data).grid("table").bind();`



<input type="checkbox"/>	eyeColor	name	company	gender
	index	email	registered	isActive
<input type="checkbox"/>	<input type="text" value="green"/>	Dean Stanley	<input type="text" value="ZENTIA"/>	male
	0	<input type="text" value="deanstanley@zentia.com"/>	<input type="text" value="2014-02-20"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="text" value="green"/>	Howard Kramer	<input type="text" value="ASSISTIX"/>	male
	1	<input type="text" value="howardkramer@assistix.com"/>	<input type="text" value="2014-03-24"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="text" value="green"/>	Grace Hardy	<input type="text" value="PROSELY"/>	female
	2	<input type="text" value="gracehardy@prosely.com"/>	<input type="text" value="2014-03-13"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="text" value="brown"/>	Rose Wilson	<input type="text" value="INTRADISK"/>	male

5. 기대효과

➤ 100% 순수 웹 표준 기술 기반

- 표준 기술 기반으로 개발을 진행하여 개발자들의 학습에 좋은 영향을 미칠 수 있습니다.
 - 비 표준 기술기반의 상용 UI 툴은 아무리 연마하고 익혀도 해당 툴을 사용하지 않는 프로젝트에서는 무용 지물이고 제조사에서의 지원이 중단되면 문제를 복구할 수 없는 상태가 됩니다.
 - 표준기술은 대부분 이전 기술과의 호환성을 유지하기 때문에 개발자들은 앞으로 등장할 웹 표준 기술들에 대해 유연하게 대처 할 수 있습니다.
- Natural-JS 는 순수 웹 표준 기술로만 이루어진 프레임워크로서 웹 개발을 지원하는 자동화 툴등을 연동하는데 전혀 제약 사항이 없고 웹 호환성과 웹 접근성을 충족하여 프로젝트의 기본적인 요구사항들을 충족 시켜 줍니다.

➤ 웹 UI 개발을 위한 통합 프레임워크 제공으로 시스템의 품질과 생산성을 동시에 증대

- 프로젝트에 Natural-JS를 도입 함으로서 웹 어플리케이션 품질의 기준이 자연스럽게 높아지고 일관성을 가지게 되며 개발생산성은 개발기간과 개발숙련도에 비례하여 큰 폭으로 증가하게 됩니다.

End of Document