

História webových riešení pre D2000

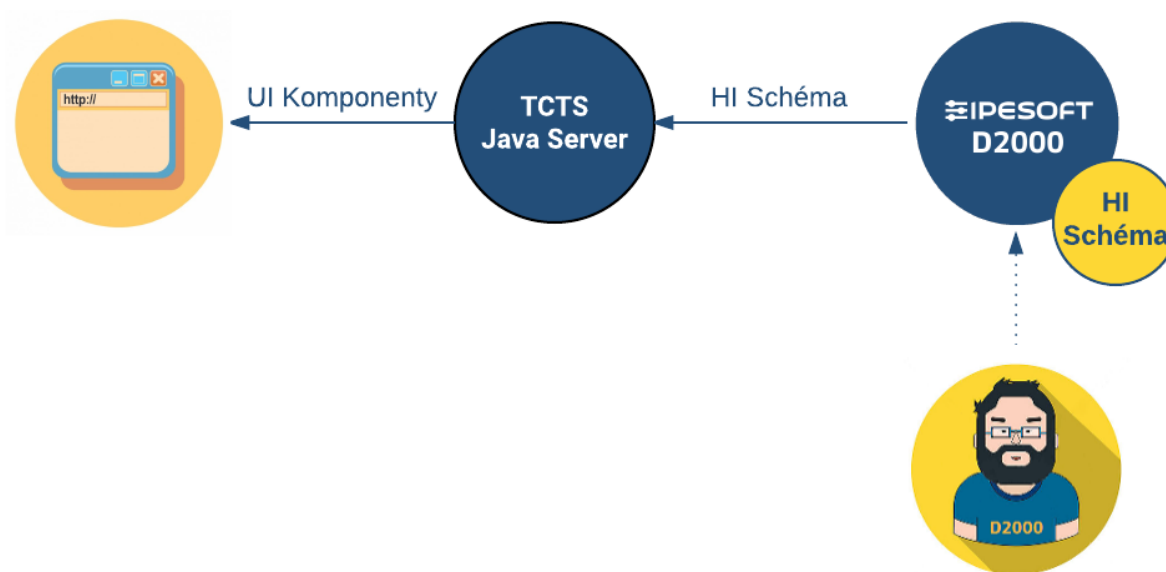
- História webových riešení pred SmartWeb platformou
 - Tenký klient
 - D2000 Websuite
- SmartWeb ako riešenie na vývoj nielen web aplikácií

História webových riešení pred SmartWeb platformou

Skôr než bude vysvetlená koncepcia samotnej platformy SmartWeb, je potrebné uviesť riešenia, ktoré SmartWebu predchádzali a realizovali viac i menej úspešne požiadavky na jednoduchý a ekonomický vývoj aplikácií.

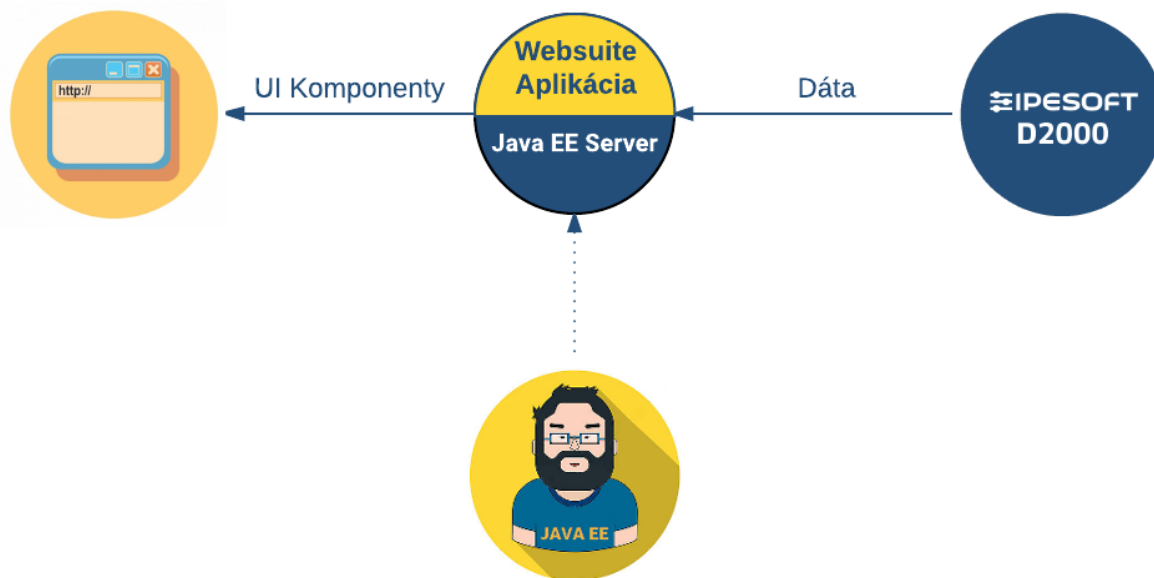
Tenký klient

Tenký klient bol prvým uceleným riešením, ktoré vyplynulo automaticky z existujúceho stavu v spoločnosti, ktorá má dostatok skúsených vývojárov na platforme D2000.



Koncept Tenkého Klienta spoíva v presnom zobrazení HI Schémy v prehliadači, teda tvorcom webovej aplikácie je samotný programátor HI schémy. Výhoda tohto riešenia spoíva predovšetkým v tom, že D2000 programátor nemusí ovládať webové technológie, postačí znalosť D2000 a o zvyšok sa postará server Tenkého Klienta (TCTS). Hlavná nevýhoda tohto riešenia vyplýva z faktu, že svet desktopových aplikácií (HI Schéma) je úplne rozdielny od sveta webových aplikácií. A preto HI schéma zobrazovaná v prehliadači sa nebude nikdy správať a vyzerať úplne identicky ako natívna webová aplikácia (napr. už len z pohľadu pozicionovania a štýlovania komponentov, využitia API prehliadača, at.). Nasadenie Tenkého Klienta má ale svoje opodstatnenie v situáciách, kedy toto obmedzenie nie je dôležité.

D2000 Websuite

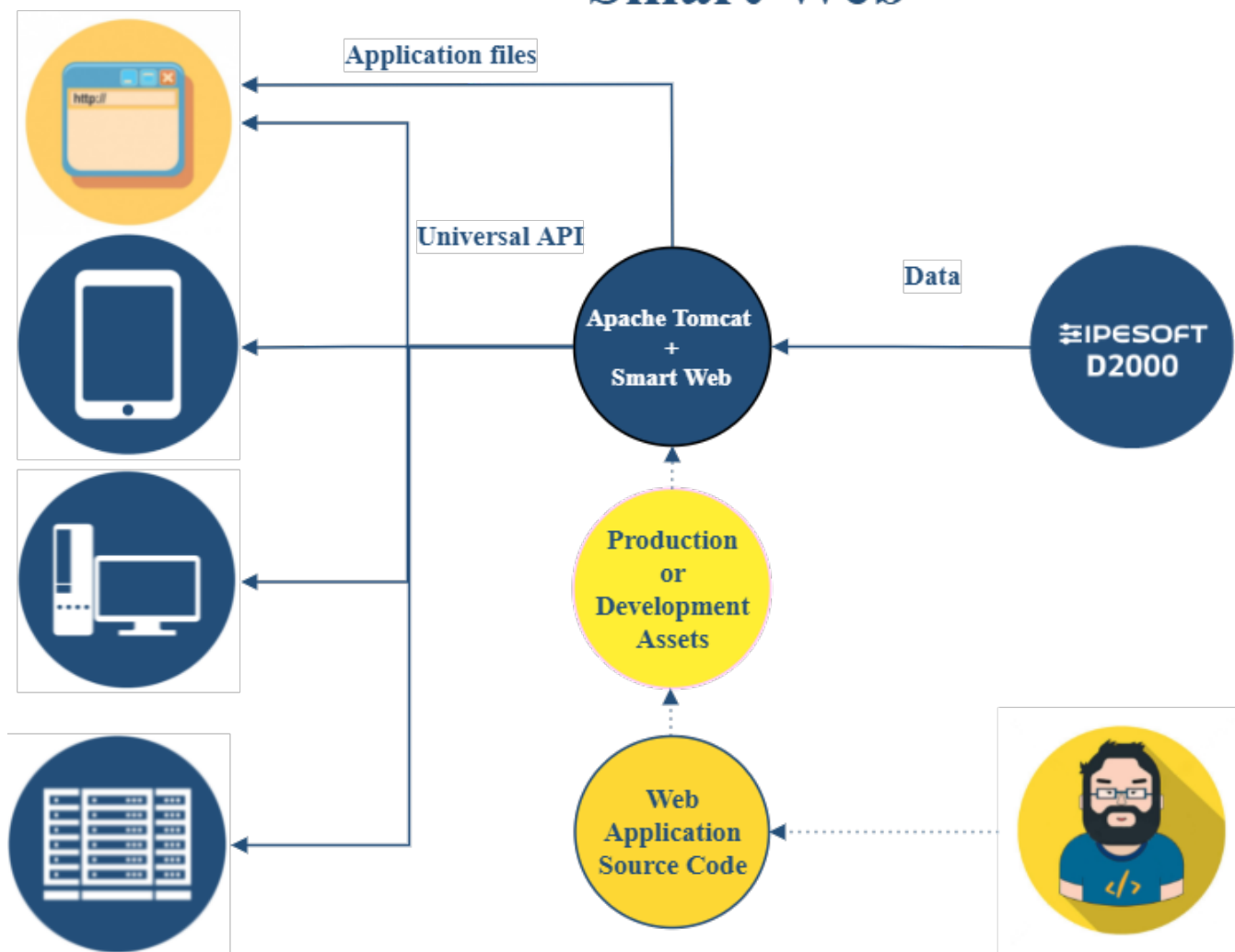


Keže Tenký Klient uspokojivo neriešil priamu realizáciu webových aplikácií, vzniklo riešenie **Websuite** postavené na aplikanom serveri implementujúcom Java EE (Enterprise Edition) špecifikácie. Java EE štandard ako taký naozaj poskytuje obrovské realizované možnosti na strane backendu (server side computing), bohužia za cenu potreby špecialistov na technológie Java EE. Websuite bol postavený na princípe predpripravených knižníc, architektúry a postupov, ktoré Java EE programátor integroval do výslednej aplikácie. Nevýhody tohto riešenia sa bohužia prejavili už pri tvorbe prvých aplikácií v podobe náronosti implementácie a požiadaviek na programátorov. Dostupných Java EE špecialistov so skúsenosami v návrhu a implementácii takýchto riešení je na trhu práce minimum a ich práca je vysoko ocenená. Z tohto dôvodu vývoj webových aplikácií založených na Websuite riešení strácal ekonomický zmysel. Druhou veľkou nevýhodou bolo použitie frontendového Java EE frameworku Java Server Faces (JSF). Sila Java EE spoíva hlavne v implementácii backendovej business logiky, ale vôbec nie v frontendových technológiách, ktoré sa vyvíjajú a menia závratným tempom. Z tohto dôvodu bolo použitie technológie JSF veľkou nevýhodou, predovšetkým ke zvažíme, že tento framework vznikol v r. 2001, o je z pohadu webových technológií úplná venos. JSF teda v konenom dôsledku skôr komplikoval vývoj webovej aplikácie, kvôli zastaralej architektúre a samotného použitia frameworku pre špecifické požiadavky klientov. Z tohto dôvodu sa aplikácie na báze Websuite riešenia už v spolonosti nevyvíjajú.

SmartWeb ako riešenie na vývoj nielen web aplikácií

Po reálnych skúsenostiach s aplikáciami postavenými na WebSuite vznikol v spolonosti dopyt na zmenu alebo vylepšenie Websuite riešenia tak, aby sa optimalizovali náklady vývoja a webových aplikácií a možnosti realizácie požiadaviek na tieto aplikácie. Po analýze možností a existujúcich skúseností s Websuite bolo rozhodnuté postaviť alší vývoj web aplikácií na novom koncepte tzv. Websuite 2.0, ktorý bol pomenovaný ako **SmartWeb**.

Smart Web



Riešenie, ktorým SmartWeb pristupuje k tvorbe webových aplikácií, je úplne rozdielne od pôvodného riešenia predstaveného v koncepte WebSuite. V prvom rade bola po analýze extrahovaná požiadavka unifikovane pristupova k dátam a službám D2000 systému z externých aplikácií - i už je to webová, desktopová, mobilná aplikácia alebo externý systém, ktorý získava dáta z D2000, alebo využíva business funkcionality implementované v D2000. Z tohto dôvodu realizuje SmartWeb **univerzálny API**, ktorým je možné pristúpi k dátam a funkcionalitám v D2000 bez ohľadu na technologickú implementáciu klienta.

Zmena nastala aj vo vývoji webových aplikácií, ktorý je oddelený od SmartWeb platformy ako takej. Spôsob vývoja frontedovej asti aplikácie môže byť teda ubovoľne prispôbený povahe projektu - zvyčajne sa používajú štandardné nástroje (tool-chain) webového vývoja, ako je napr. npm, yarn a webpack. Vysledné produkčné alebo vývojové assety (html, css, js) SmartWeb hostuje klientským prehliadačom z konfigurovaných ciest v súborovom systéme. Smart Web umožňuje poskytovať viacero verzií tej istej aplikácie naraz prostredníctvom iných URL ciest, t.j. jedna Smart Web inštancia môže naraz napr. servovať ako produkčnú verziu web aplikácie tak aj niekoľko vývojových verzií. SmartWeb server **poskytuje dotatone funkcionality na podporu nasadenia** takýchto webových aplikácií. Jedná sa napr. o spoločnú autentifikáciu, monitoring, logovanie, at.

Treťou pridanou hodnotou SmartWeb platformy sú **javascript/typescript knižnice** publikované v npm repozitári využívané pre rýchlu a efektívnu implementáciu webovej aplikácie nad D2000. K dispozícii sú aj **zdrojové kódy demo aplikácie** pre rýchly štart vývoja aplikácie pre Smart Web.

Všetky spomínané vlastnosti SmartWeb servera sú popísané v nasledujúcich kapitolách.