Treball Final de Màster, PAC2

Alejandro Exojo

Introducció

- WordPress: programari per construir llocs web amb gran popularitat
- Fet servir abastament per particulars i empreses de mida petita a gran
- Eina de comunicació en la web de molta importància
- Necessitat d'identificar els interlocutors amb fiabilitat
- Conclusió: estendre WordPress amb un plugin que permeti la signatura de comentaris i d'entrades («posts»)

Objectius

- Introduir-se al desenvolupament de plugins de WordPress
- Aprendre els conceptes de la signatura mitjançant X.509
- Desenvolupar una petita biblioteca per signar amb l'objecte «crypto» del costat del client
- Finalitzar un plugin per a WordPress que faci servir la biblioteca implementada

Antecedents

- No existeixen antecedents de cap programari similar per a WordPress
- En altres gestors de continguts de l'estil (Drupal, Joomla!) tampoc

Antecedents

- No existeixen antecedents de cap programari similar per a WordPress
- En altres gestors de continguts de l'estil (Drupal, Joomla!) tampoc

Explicació del projecte (1)

- WordPress: programari per gestió de weblogs, amb capacitats de gestor de continguts
- Basat en PHP i MySQL
- Altament extensible via plugins
- Us freqüent de jQuery per millorar la experiència d'usuari al costat client
- La publicació als blogs és eminentment persona a persona
- Punt d'interès: verificació/no repudi del contingut al blog

Explicació del projecte (2)

- X.509: estàndard de infraestructura de clau pública
- Especifica certificats, claus públiques/privades, llistes de revocació, etc.
- PKCS#12: format que emmagatzema diferents arxius necessaris per a X.509
- Està suportat per Mozilla per a certificats personals
- Els navegadors de Mozilla (FireFox/SeaMonkey) suporten l'API crypto per a signar amb certificats personals

Explicació del projecte (3)

- Conclusió: integrar aquestes tecnologies
- Creació d'un plugin de WordPress
- L'usuari es registra, i carrega la seva clau pública
- Signatura feta al client (navegador) amb JavaScript i la clau privada
- Text signat verificat pel servidor (WordPress) amb la clau pública, i mostrada a la resta

Viabilitat

- Us de tecnologies molt arrelades a la comunitat: WordPress, Mozilla FireFox, etc.
- Facilitat d'us: només l'administrador del web ha d'instalar programari
- Els usuaris existents poden verificar l'autoria sense cap intervenció

Metodologia

- Ús de la documentació existent en la comunitat de programari lliure
- Programari enterament lliure: PHP, MySQL, WordPress, FireFox + FireBug, OpenSSL / GNU TLS
- Us actiu de les llistes de correu i mecanismes de comunicació ja establerts
- Us dels sistemes de control de versió de cada projecte

Cost

- Familiarització amb WordPress: 2 setmanes
- Us bàsic de la signatura al client en JavaScript: 2 setmanes
- Biblioteca JavaScript per la signatura: 2 setmanes
- Desenvolupament del plugin WordPress: 2 setmanes
- Integració del JavaScript al plugin: 1 setmana
- Proves: 2 setmanes
- Comparació de rendiment: 1 setmana
- Total: 12 setmanes (12 per aproximadament 30 hores setmanals), 360 hores

Novetats

- Inexistència de plugins semblants a WordPress
- El projecte cobrirà un nínxol que estava buit fins a la data

Implementació: maqueta inicial

- Comencem amb un plugin simple que funcioni
- El plugin no fa cap funció encara
- Però afegeix interfícies als comentaris i a la edició
- Les parts fonamentals són al seu lloc

Implementació: entorn de signatura en el client

- Passem a interactuar amb OpenSSL
- Creem una Autoritat de Certificació
- Signem un certificat amb aquesta A.C.
- Formem un arxiu PKCS#12 i l'importem al navegador
- El navegador està llest per a signar

Implementació: prova de concepte

- Programem un programa del costat del client
- Llegeix un formulari, el signa, i el guarda en altre camp
- El sistema està llest per a enviar les dues parts al servidor

Implementació: verificació al servidor

- El servidor rep dos camps: missatge i signatura
- Provem la funció openssl_pkcs7_verify()
- Requereix desar els continguts al disc
- Requereix un missatge S/MIME
- Funció a poc alt nivell, i amb massa complicació
- Invoquem openssl directament
- Els components es deixen al disc igualment

Implementació: integració al plugin definitiu

- La signatura funciona, el plugin funciona també: queda unir les dues parts
- Afegim més interfícies: definir una Autoritat de Certificació del lloc web
- Afegim un script del costat del client: genera la signatura en el clic pertinent
- Afegim les funcions de verificació al plugin
- Desem la signatura a la base de dades com a metadades del post o comentari

Conclusions

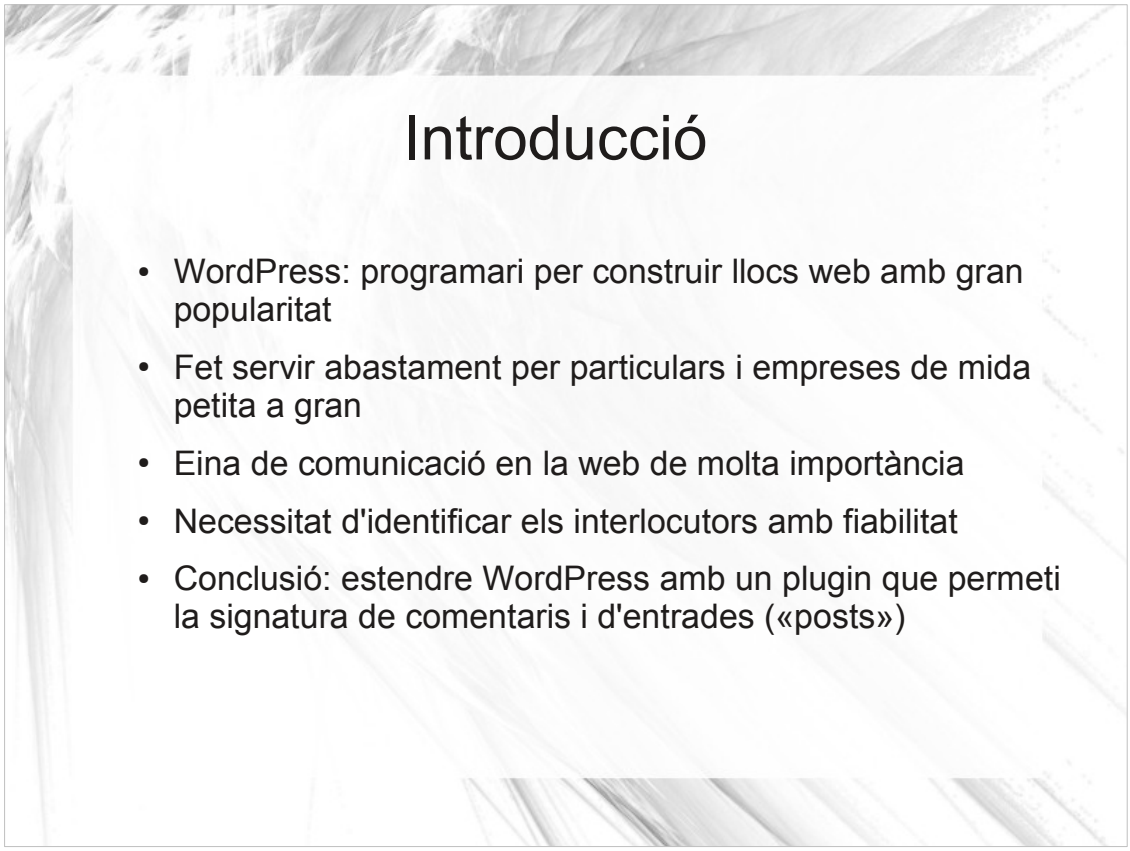
- Utilitat del projecte clara
- Casos d'us garantits
- Implementació de cada funcionalitat en parcel·les ben diferenciades
- Us de tecnologies madures i ben assentades
- La consecució dels objectius semblava una fita raonable, i s'han assolit

Ampliacions o millores

- Interfície i presentació principalment
- L'usuari que llegeix ha de tenir clar el missatge
- Obtenir la signatura i el missatge en cru per a la seva verificació manual

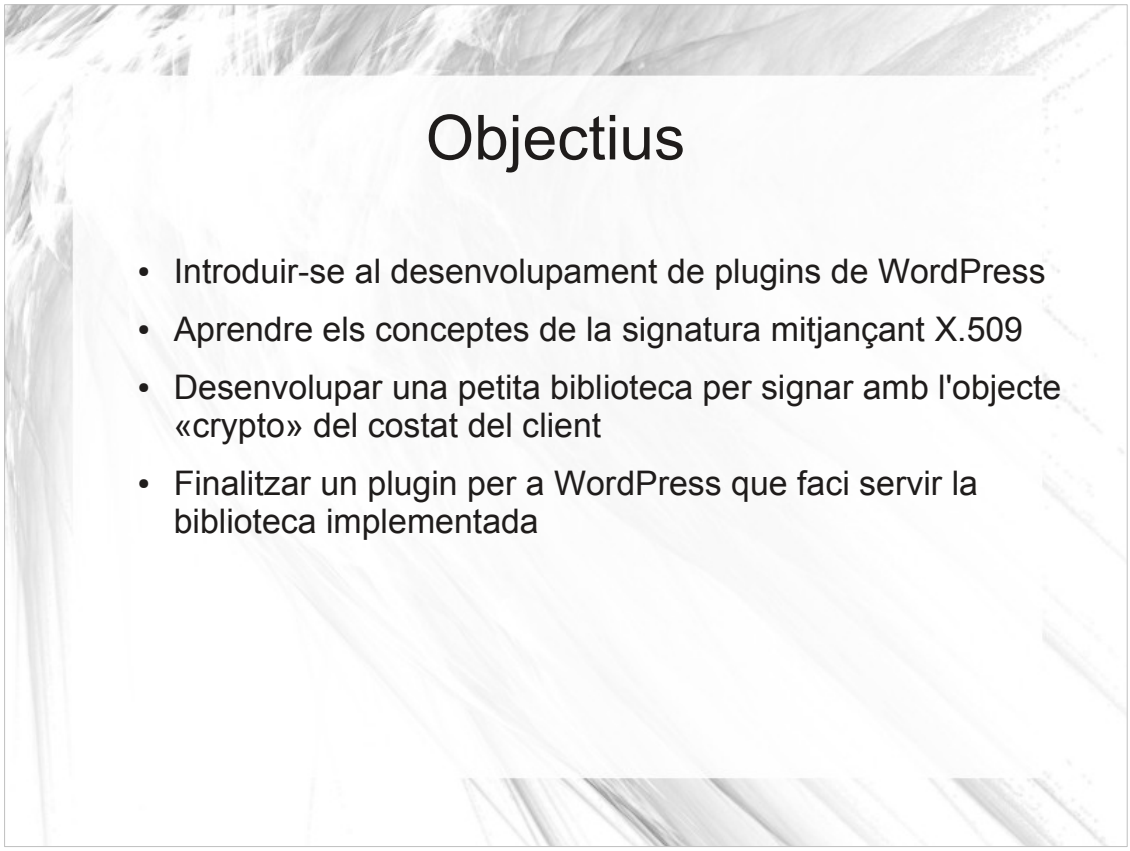
Treball Final de Màster, PAC2

Alejandro Exojo

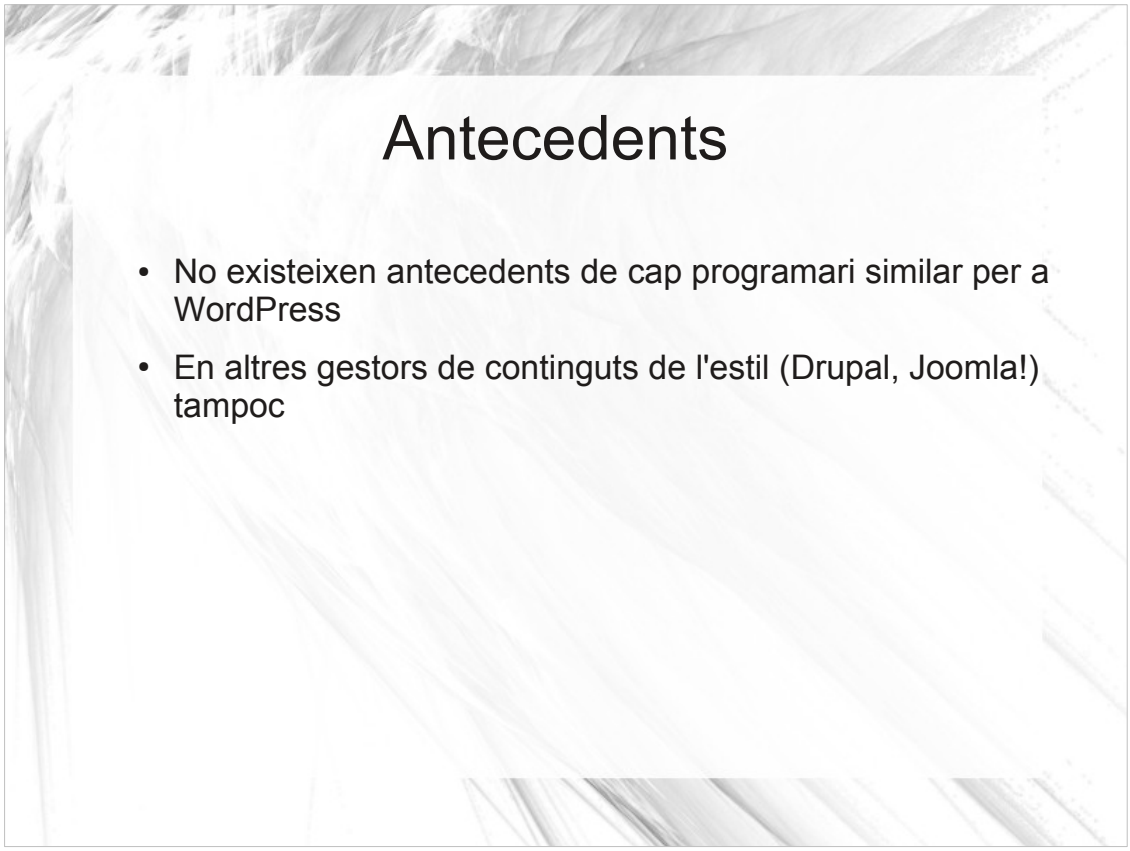


WordPress fa temps que s'ha assentat com un programari molt popular per construir no només weblogs, sino també altres llocs web. És fet servir per empreses de moltes mides diferents.

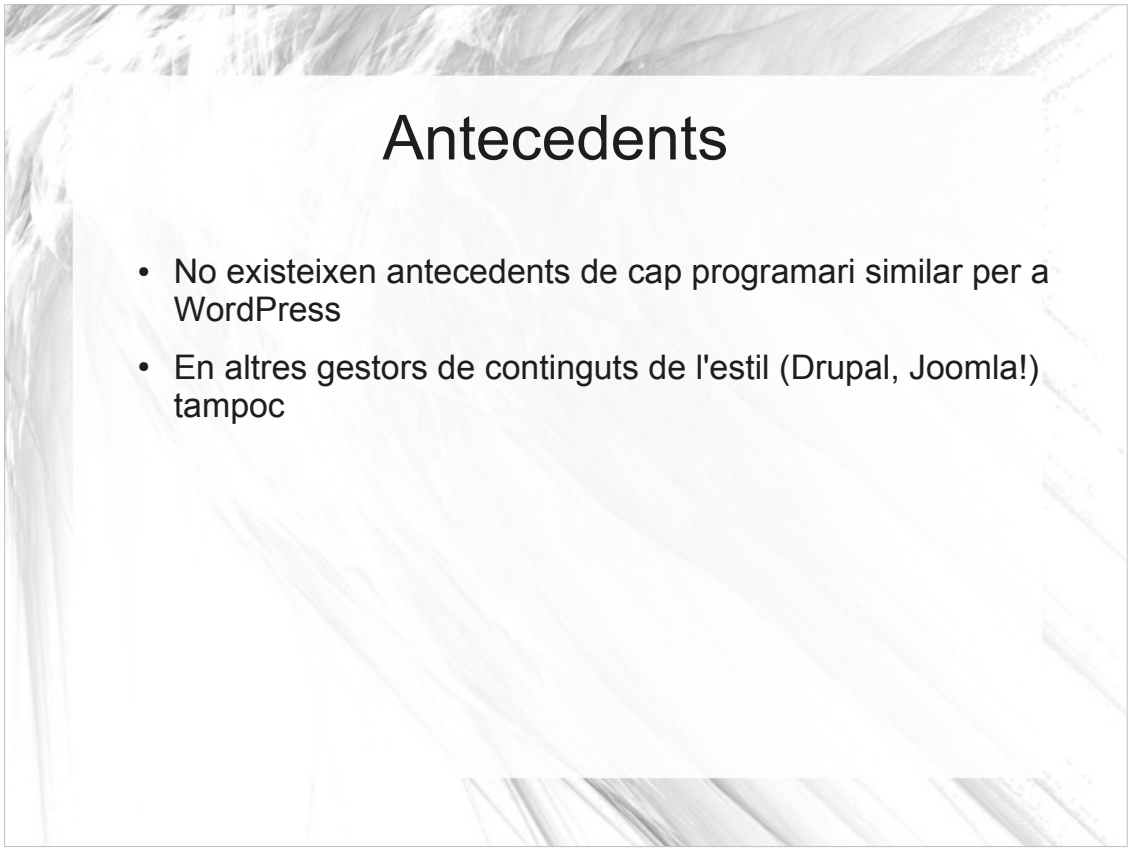
Resulta doncs una eina de comunicació força important, i en aquesta comunicació és important poder disposar de la possibilitat d'identificar amb fiabilitat l'autoria d'un missatge.



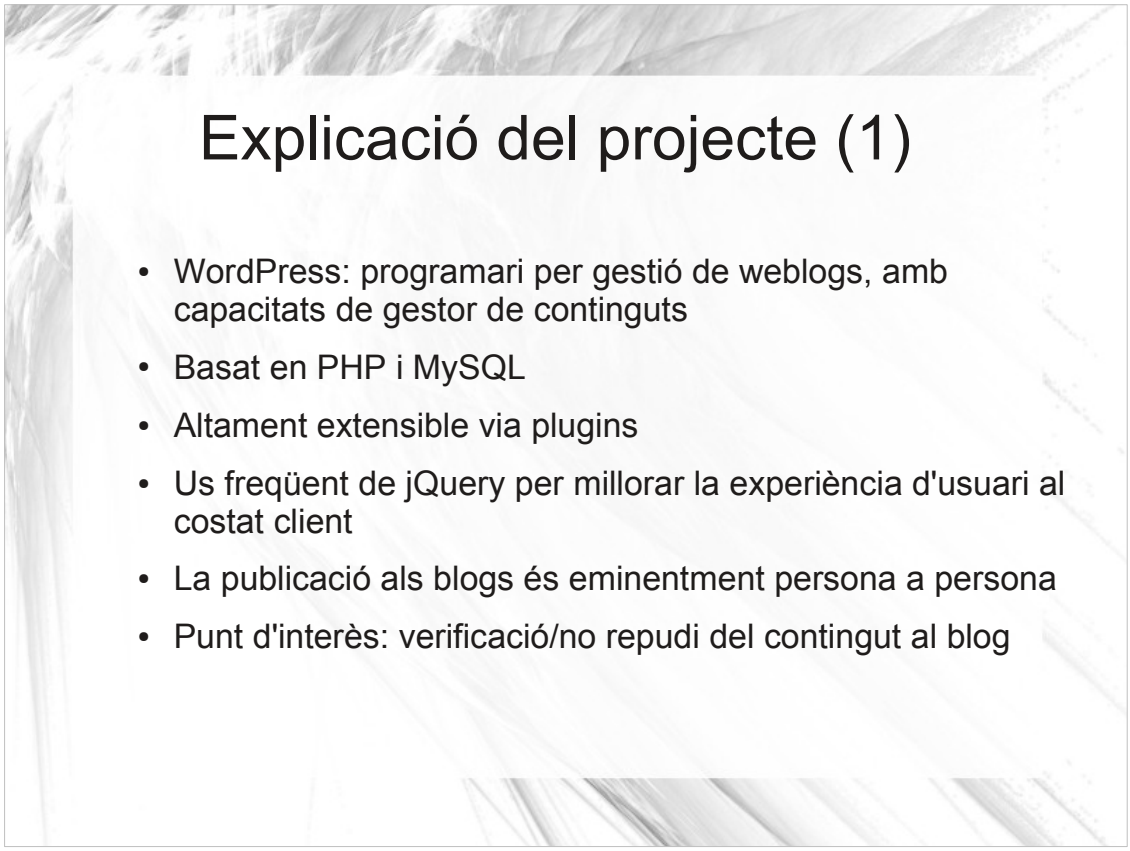
Es començarà per aprendre el desenvolupament de extensions per al programari WordPress.
S'aprendrà també aquells coneixements de la infraestructura que defineix X.509 per a la signatura electrònica de persones.
Es desenvoluparà el codi necessari en JavaScript per fer servir l'objecte crypto dels navegadors basats en Gecko.
Tot s'integrarà en un plugin de WordPress.



No s'ha trobat altre programari equivalent ni per WordPress ni per altres sistemes gestors de continguts populars.



No s'ha trobat altre programari equivalent ni per WordPress ni per altres sistemes gestors de continguts populars.



WordPress és un programari per a la creació de weblogs amb moltes funcionalitats. Ha guanyat capacitat per a gestionar altres continguts en les darreres versions.

Està programat en PHP, i fa servir MySQL per les dades.

És altament extensible via plugins, i està preparat per un us important de codi del costat del client.

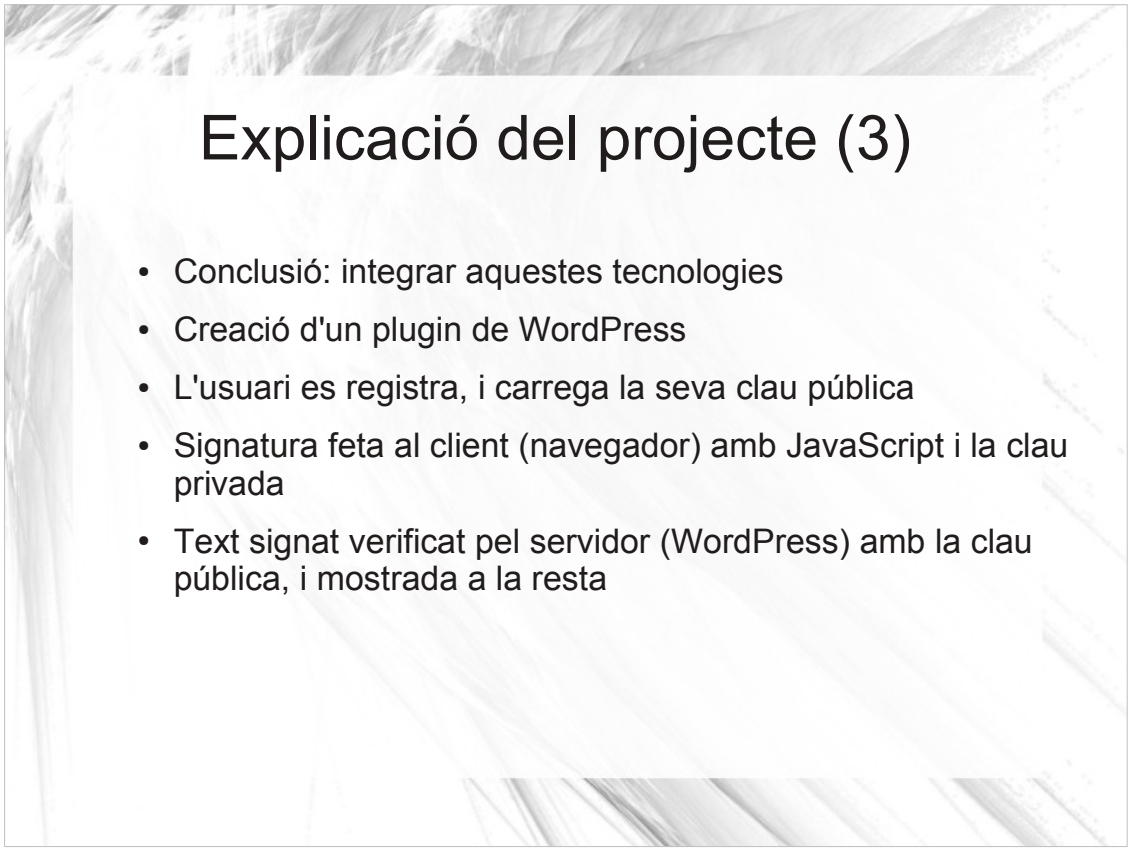
Com que la comunicació als blogs és majorment de persona a persona, és cada vegada més important identificar als interlocutors del missatge.

Explicació del projecte (2)

- X.509: estàndard de infraestructura de clau pública
- Especifica certificats, claus públiques/privades, llistes de revocació, etc.
- PKCS#12: format que emmagatzema diferents arxius necessaris per a X.509
- Està suportat per Mozilla per a certificats personals
- Els navegadors de Mozilla (FireFox/SeaMonkey) suporten l'API crypto per a signar amb certificats personals

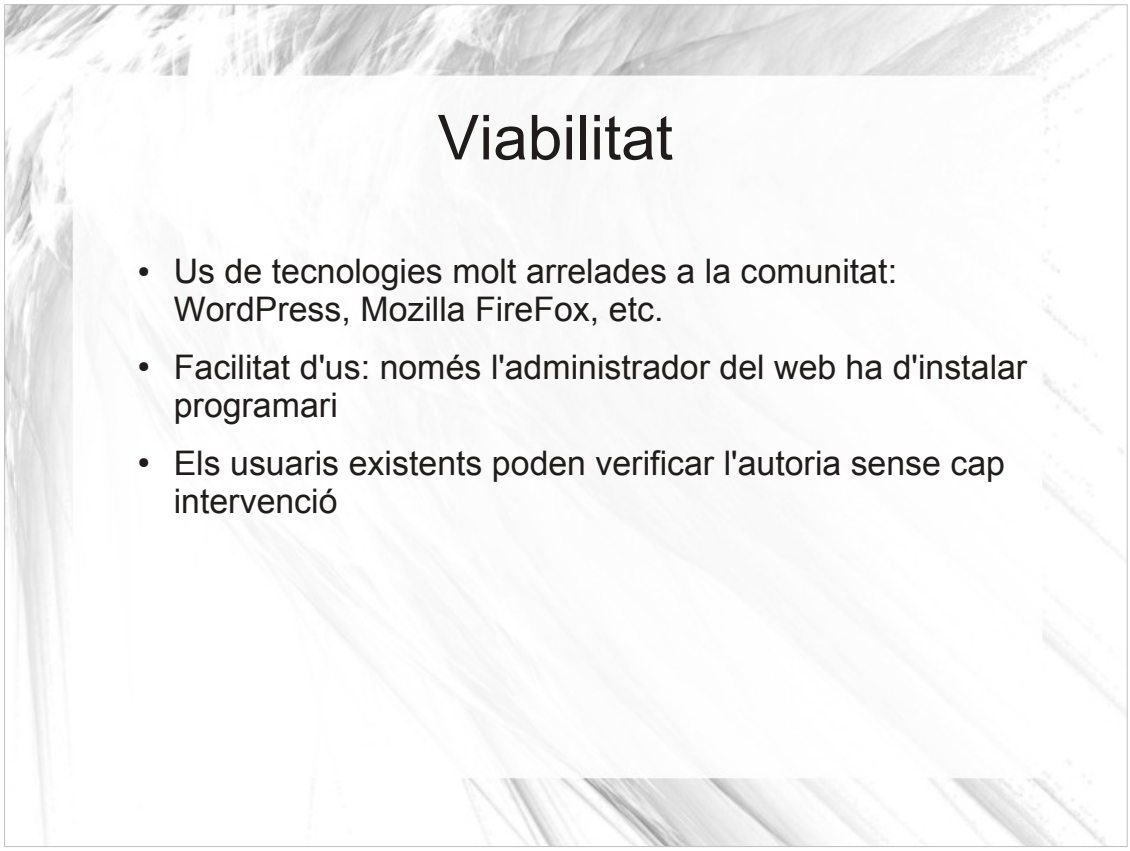
X.509 és un estàndard d'infraestructura de clau pública, i el format PKCS#12 és el que empra FireFox per a importar els certificats de l'usuari al navegador.

Els navegadors basats en el motor Gecko (família Mozilla), inclouen una API per a emprar els certificats des de JavaScript.



Per tant, resulta clar l'interès d'integrar aquests possibilitats. Es crearà un plugin de WordPress que verificarà la signatura de comentaris i entrades fets per usuaris que s'hagin registrat i hagin pujat la seva clau.

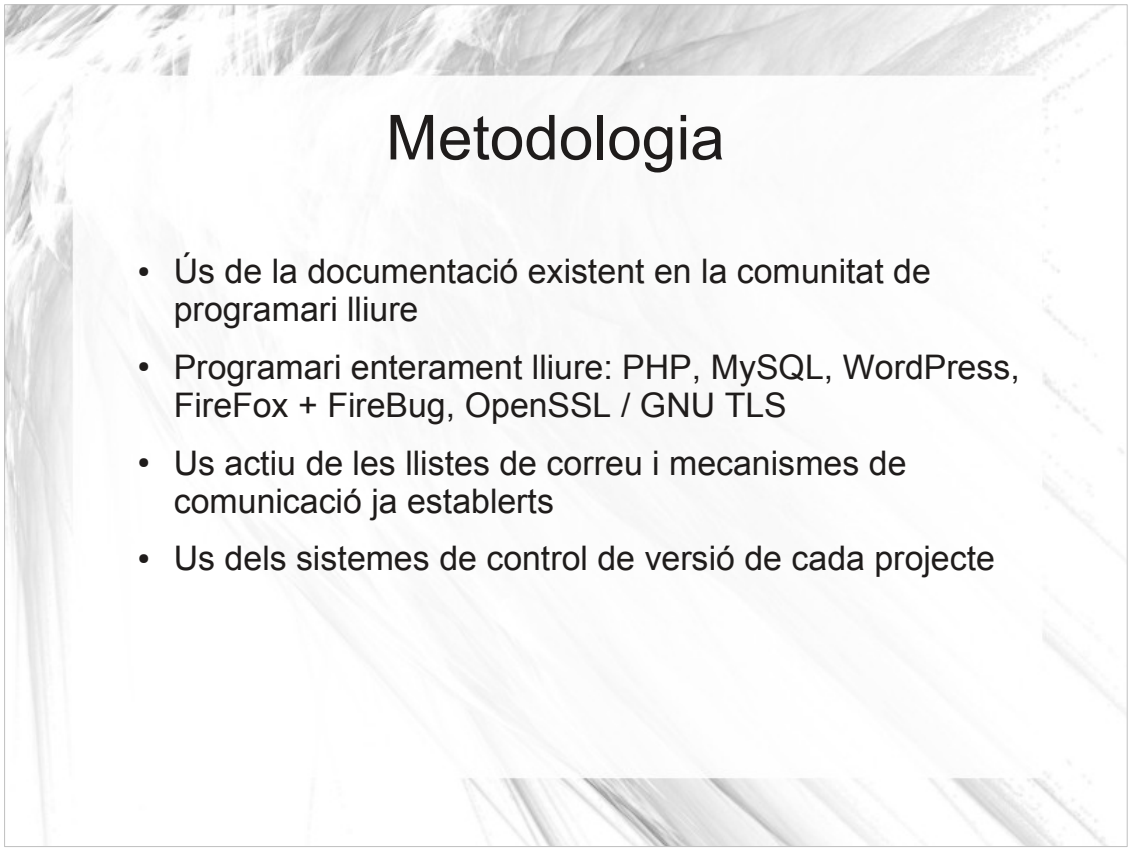
El contingut es signa del costat del client (navegador) amb JavaScript, a on roman la clau privada.



La viabilitat i la utilitat del projecte són favorables. Totes les tecnologies emprades gosen de popularitat i maduresa.

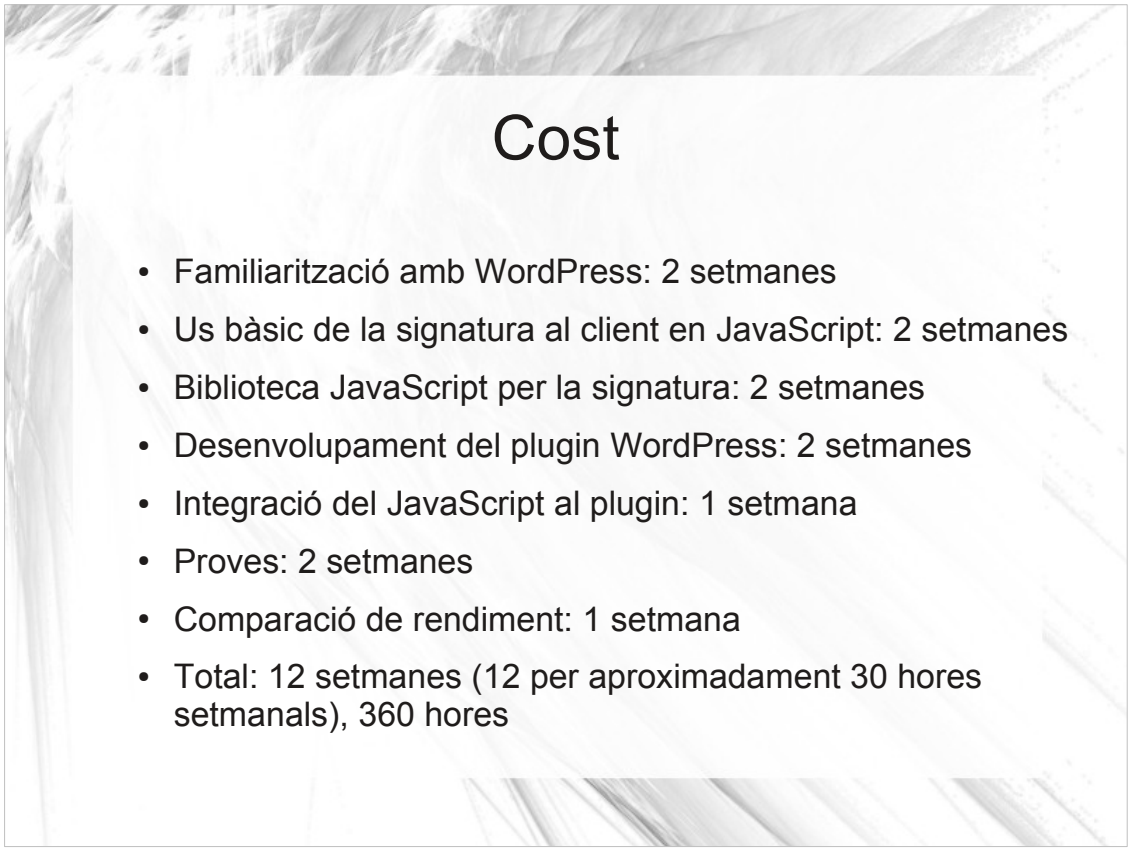
Aquesta comunitat al darrere garanteix alta probabilitat d'obtenir massa crítica d'usuaris.

Els usuaris no registrats poden gaudir de la funcionalitat del plugin (verificació del contingut) sense fer res. Això facilita la implantació i la utilitat enormement.

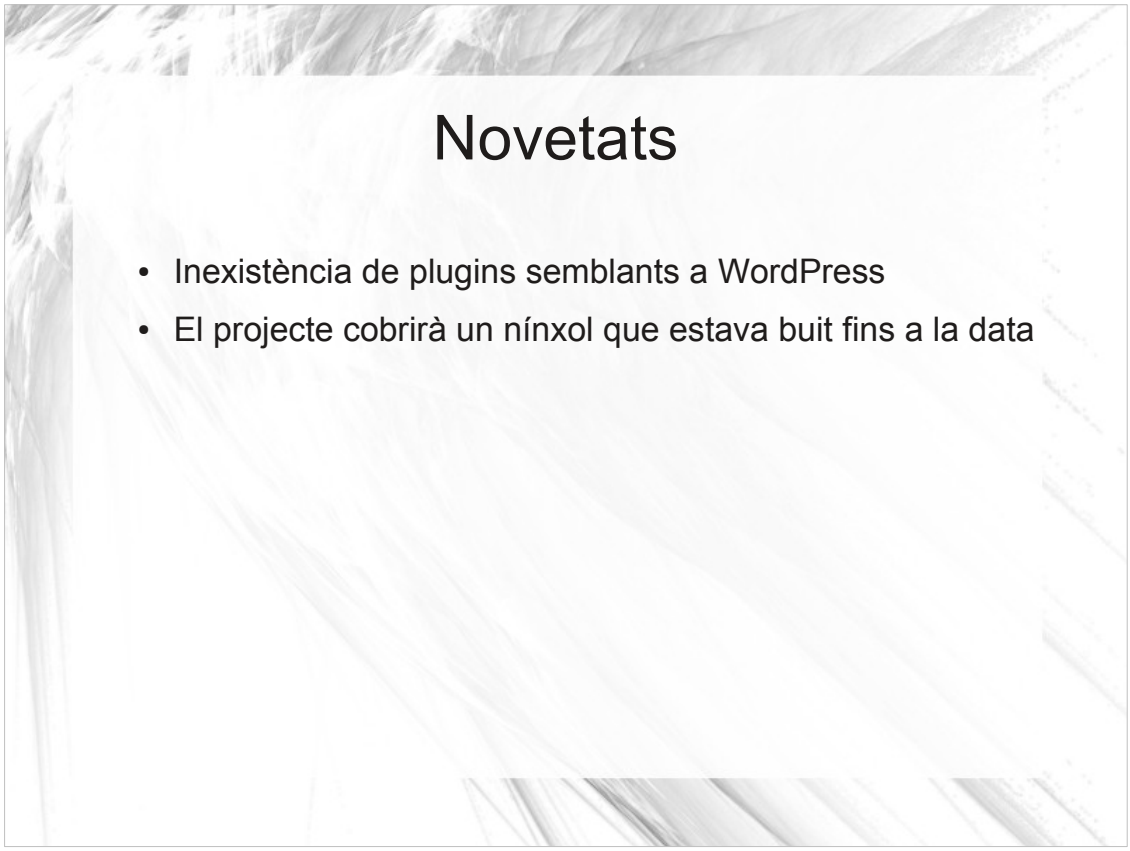


Es farà servir el programari lliure habitual, especialment aquell propi de les comunitats implicades: eines lliures de desenvolupament, control de versions, comunicació, i depuració.

Es farà èmfasi en comunicar correctament l'existència del projecte a les parts interessades en la comunitat de programari lliure.

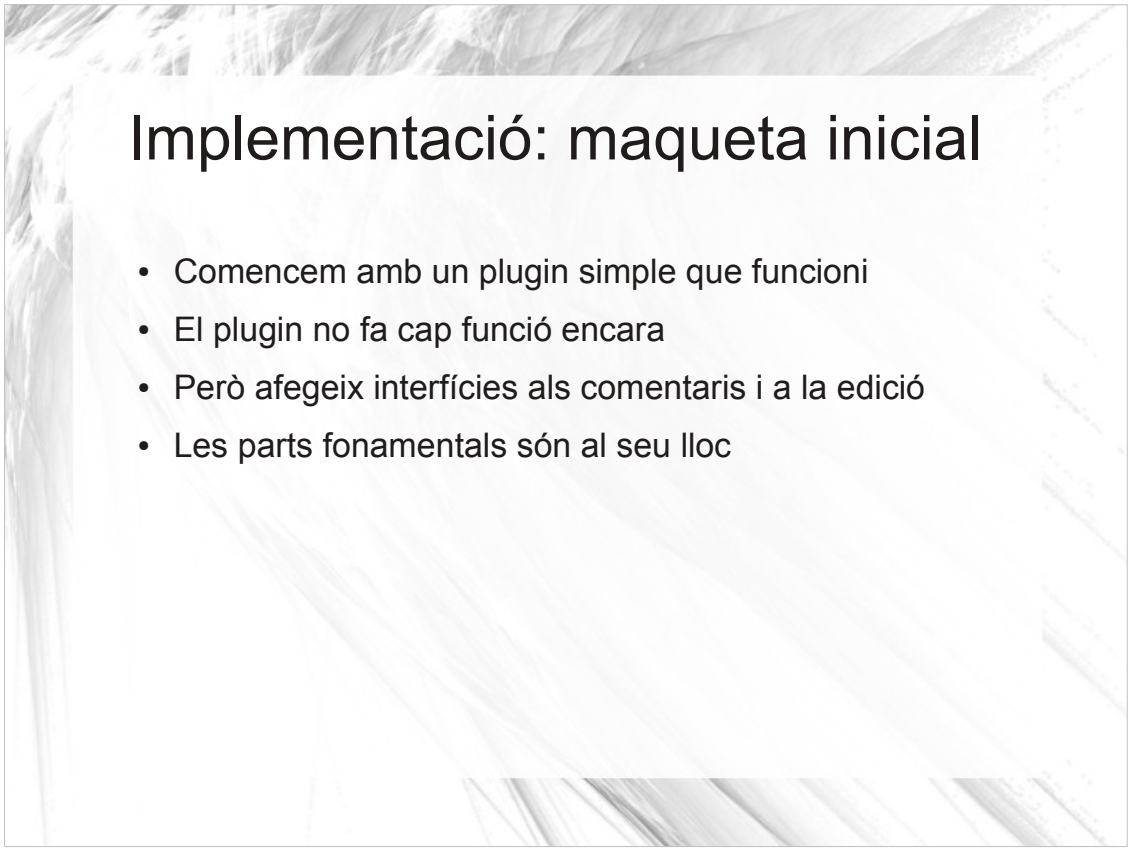


La divisió en tasques és la suggerida pel consultor de l'empresa. L'assignació de temps és decidida per l'alumne i consensuada amb el consultor.

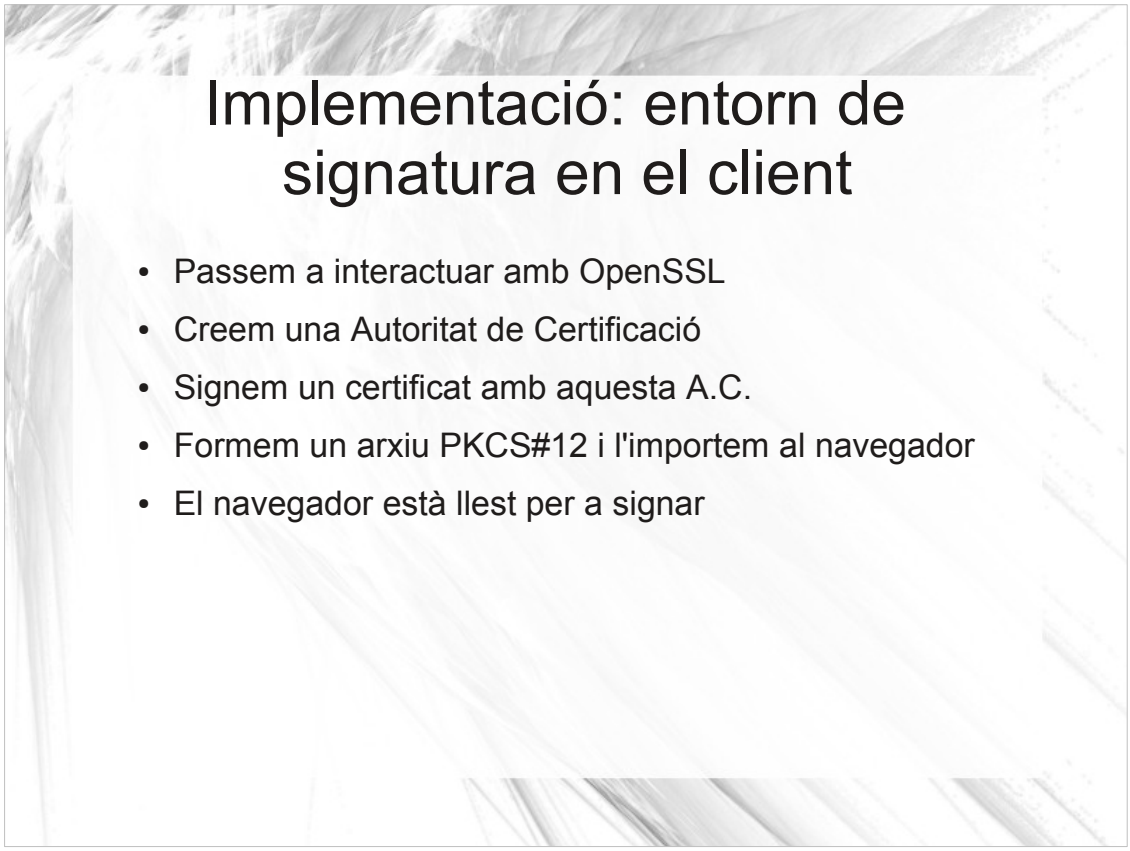


Com s'ha mencionat, no s'han trobat altres extensions que facin servir aquesta funcionalitat. Probablement el fet que l'API crypto sigui només implementada per Mozilla, fa que no sigui especialment popular.

Es considera que el projecte cobrirà una necessitat que fins ara no tenia resposta.



Comencem creant l'esquelet més bàsic d'un plugin de WordPress que sigui la mínima expressió.
Després, comencem a actuar sobre ell, estenent la interfície d'usuari de WordPress a on caldrà que hi sigui en el producte definitiu.
Comencem per afegir una caixa amb una casella a la interfície d'edició.
Seguim afegint la funcionalitat per a que es guardi l'estat de la casella a les metadades de la pàgina editada.
Es repeteix el procediment per als comentaris.
Amb aquestes funcions al seu lloc, serà una qüestió de estendre-les en fases posteriors.

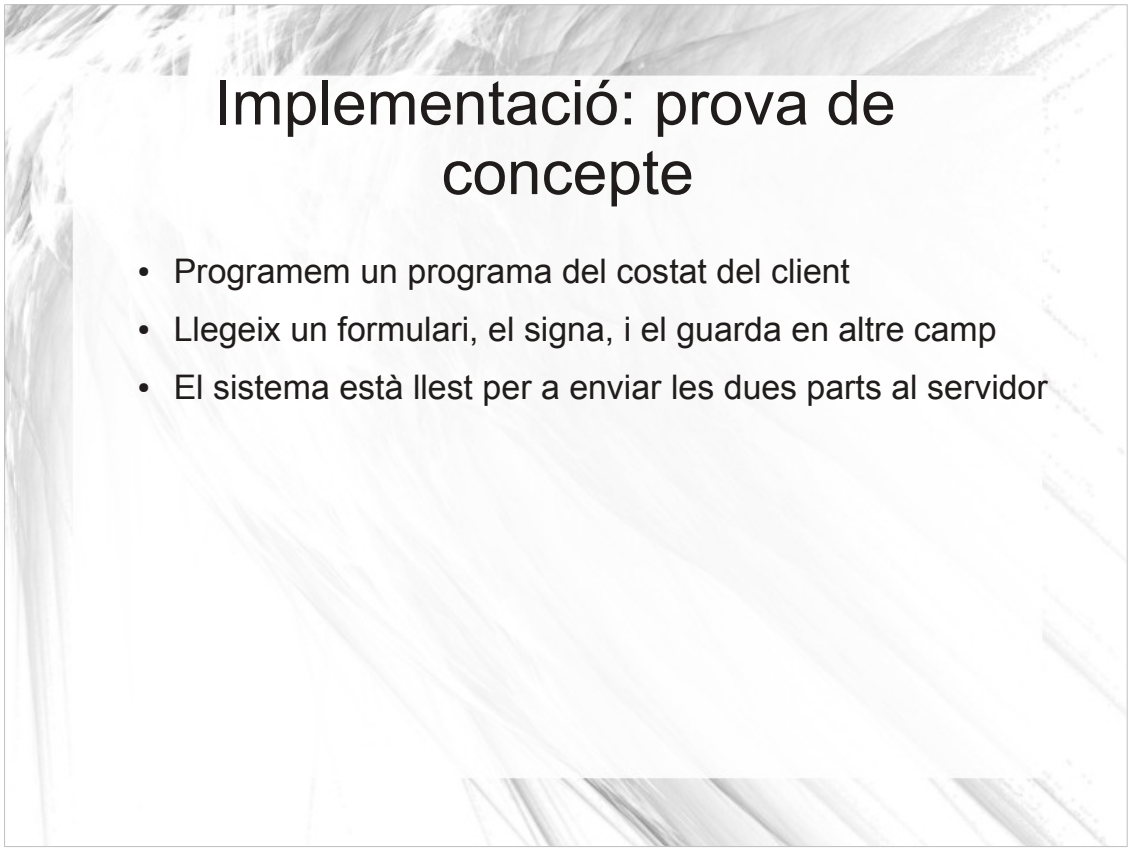


Creem una autoritat de certificació amb una clau autosignada.

Aquesta autoritat ja pot signar altres certificats que correspondran a usuaris ficticis.

Això ens permet tenir l'entorn de proves suficient.

Cal generar arxius “.p12” i importar-los al navegador, juntament amb l'autoritat de certificació, per a que el navegador pugui signar.



Creem un petit script del costat del client (en JavaScript).

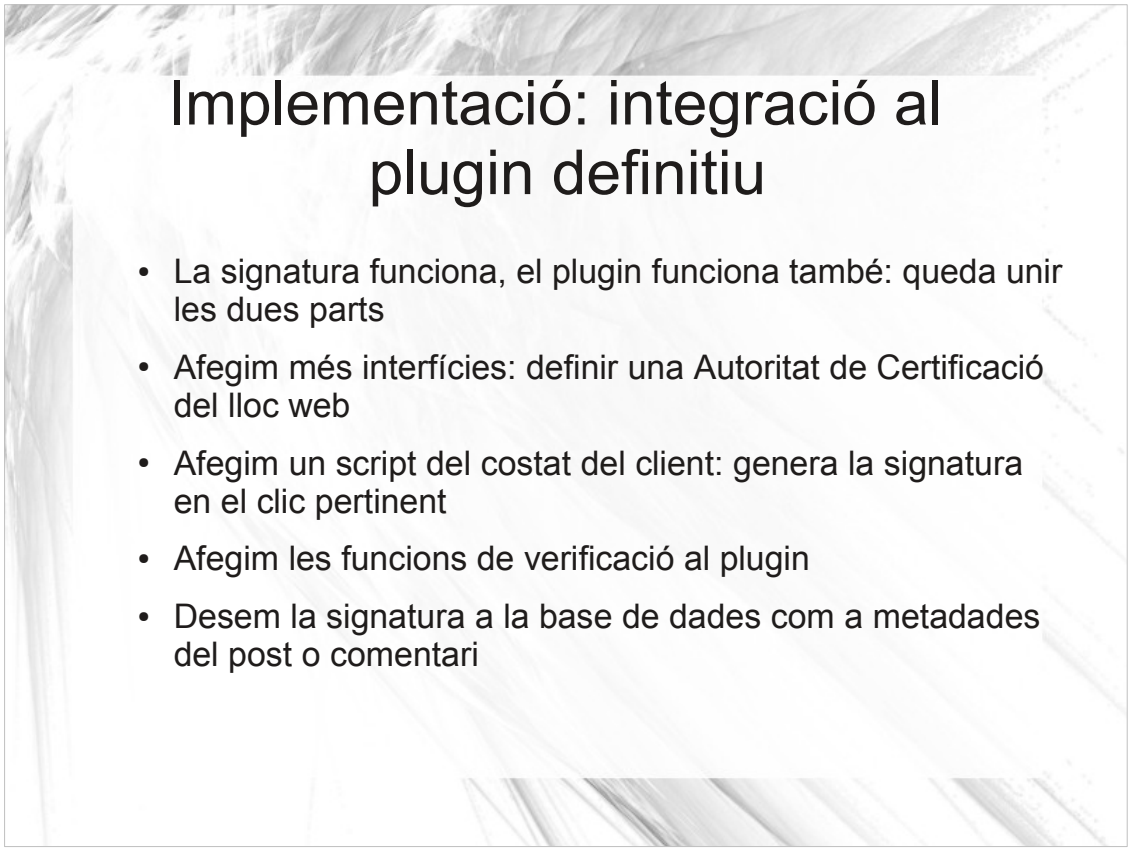
El programa s'invocarà en fer clic al formulari, i llegirà el contingut del formulari, a on hi haurà un suposat text que l'usuari vol signar, i invocarà la funció signText() per a que l'usuari el signi.

Enmagatzema el resultat en un altre camp, i ho deixa enllestit per a enviar-ho al servidor.

Implementació: verificació al servidor

- El servidor rep dos camps: missatge i signatura
- Provem la funció openssl_pkcs7_verify()
- Requereix desar els continguts al disc
- Requereix un missatge S/MIME
- Funció a poc alt nivell, i amb massa complicació
- Invoquem openssl directament
- Els components es deixen al disc igualment

El servidor rep via POST el contingut del formulari.
Rep dues variables separades, el missatge, i la signatura.
Existeix un API de PHP per interactuar amb OpenSSL, però és immadura, poc documentada, i de baix nivell.
Requereix desar els missatges al disc, ja que no treballa amb variables en memòria. A més, requereix composar un missatge en S/MIME.
Fem servir les nostres pròpies funcions per cridar a OpenSSL directament.



Finalment, integrem les diferents parts.

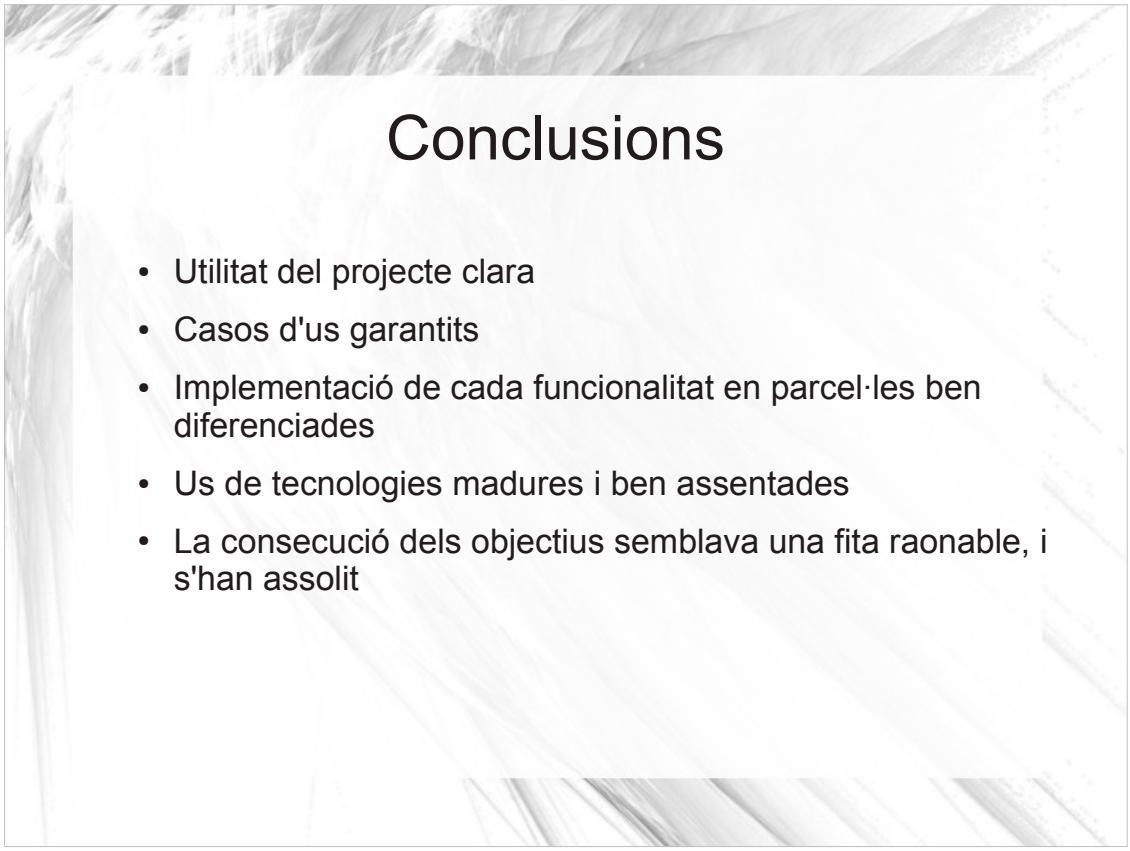
Cal afegir una nova interfície, la que permet configurar al lloc web una autoritat de certificació pròpia.

És molt convenient per a fer proves (sense haver de pagar per certificats signats per autoritats arrel), però també resulta molt útil per a intranets o organitzacions o institucions que poden expedir els seus propis certificats. Exemple: la FNMT no sempre és reconeguda.

Creem un codi JavaScript nou que es registra a l'event onClick de la interfície de WordPress.

Afegim la signatura generada a la base de dades com a metadades del post o comentari que s'està creant o editant.

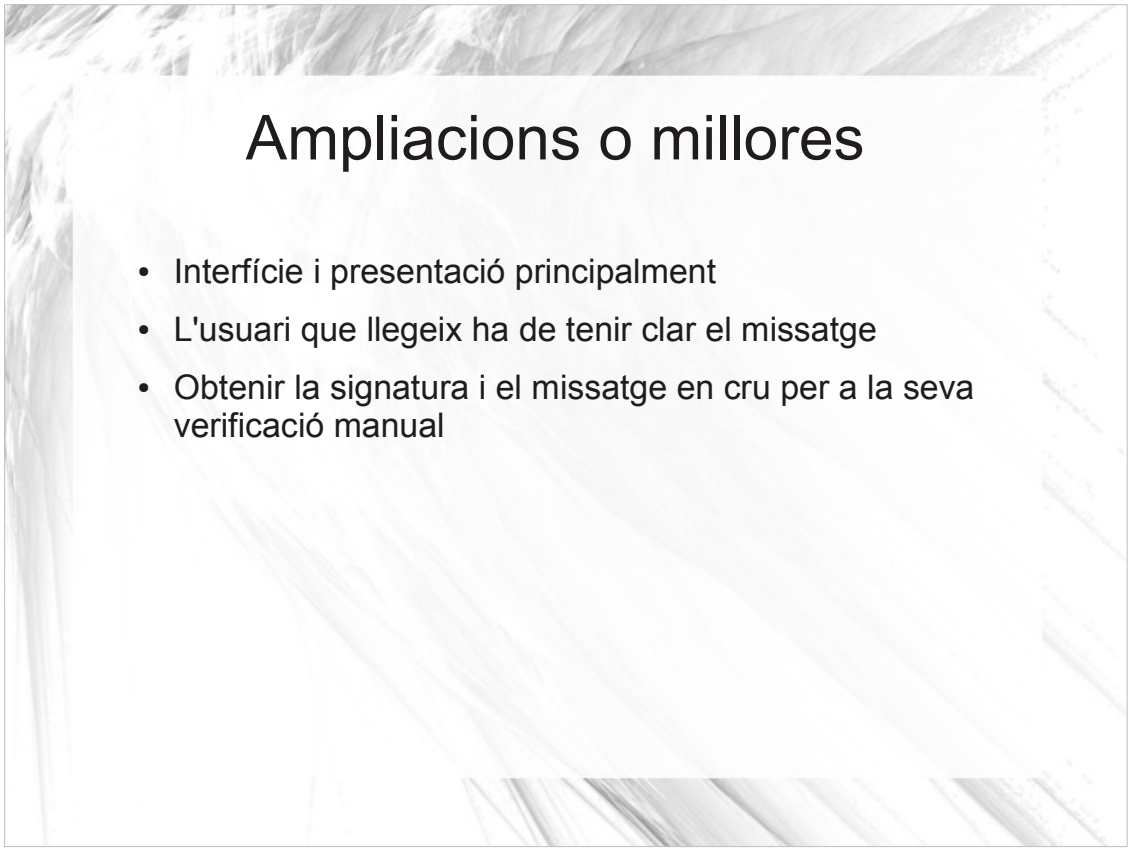
No requereix instal·lació de taules noves a la BBDD.



Podem dir que el projecte té una utilitat clara, i que és força probable que arribi a tenir un ús final més enllà de l'acadèmic.

Cada part de la funcionalitat s'executa en àmbits diferents, per tant es simplifica l'arquitectura en quant a la mida de les parts.

Es considera que aconseguir els objectius és una fita perfectament possible en els temps establerts.



La interfície és millorable, especialment l'aspecte.

Requeriria de «feedback» amb usuaris reals en entorns reals.

Possibilitat de permetre descarregar el contingut en cru del comentari (sense filtrar per WP), així com la firma, per a la verificació manual sense haver de confiar en el lloc web.