BSV Browser: Risk Review, Unanswered Questions and Corrective Recommendations

Applying BRC Opinions 151 and 152 to BSV Association's Browser-Wallet Gateway

Documentation and Source-Code Review as of August 6, 2026

Table of Contents

Executive Finding

BSV Browser is not simply a web browser with a Bitcoin wallet attached.

It brings a browser, self-custodial wallet, persistent cryptographic identity, identity certificates, application permissions, transaction history, payment routing, wallet-state storage and recovery into one application. Compatible websites operating inside the browser can ask the wallet to create transactions, provide keys, sign or encrypt information, reveal certificate fields and perform other operations through the BRC-100 wallet interface.

That is an impressive amount of power to place behind one icon on a phone.

It is also an enormous amount of risk to place behind one icon on a phone.

- A normal browser failure may expose browsing information.
- A wallet failure may threaten money.
- An identity-system failure may expose credentials.
- A permission failure may let an application do something the user never intended.
- A recovery failure may leave the user holding the private keys but unable to reconstruct the records needed to find, organize or use everything associated with them.

BSV Browser brings all of those risks together.

This paper does **not** claim that BSV Browser has been proven to steal funds, spy on its users or deliberately create a surveillance system. It has not.

The concern is more basic and, in some ways, more important:

> A system built to free users from gatekeepers can quietly create a new and more powerful gatekeeper if too much money, identity, history, permission and recovery authority is allowed to collect inside one wallet and around the services chosen for it.

BRC Opinion 151 warned that BRC-100 could become a common gateway for money, identity, applications and autonomous software. It did not allege that every capability was already being abused. It asked whether the architecture could become vulnerable to surveillance, exclusion, legal pressure, infrastructure failure or loss of independent user control.

BRC Opinion 152 addressed a related danger: placing identity-verified or regulated activity inside the same wallet environment as activity the user expects to remain pseudonymous. Its central warning was simple:

> **A separate basket is not a real identity firewall. A separate wallet is.**

BSV Browser is the first major real-world stress test of those warnings.

Some findings are positive. The product is publicly source-available. New wallets are described as using local SQLite storage and no Wallet Authentication Backend, or WAB. Users retain their private

keys. The wallet offers permission screens, conventional address payments through its Legacy Bridge, database export, printable recovery shares and user-selectable ARC transaction-processing endpoints.

But the same review found serious problems and unanswered questions:

- Identity-based payment messaging points by default to `messagebox.babbage.systems`.
- Ordinary users are not plainly told what that service does, what it can observe or how to choose another provider.
- The public production-build configuration contains hosted WAB and storage endpoints even though the current source and README describe no WAB and local storage as the hardcoded defaults.
- The support guide repeatedly promises that every payment requires explicit approval, while the source snapshot reviewed sets an automatic-approval threshold of 100,000 satoshis with a ten-second cooldown.
- The recovery phrase does not reconstruct the complete wallet state under the documented recovery process without a separate wallet-database export.
- Complete recovery into a genuinely independent wallet has not been demonstrated.
- No first-class separation has been demonstrated between savings, normal spending, AI-agent funds and regulated or identity-linked assets.
- Several critical protections remain undocumented or impossible to verify without testing the actual released application.

These are not small details.

Financial and identity infrastructure cannot ethically operate on the quiet assumption that ordinary people are willing beta testers. Developers may understand that a product is experimental, changing quickly and dependent upon real-world testing. Most users do not.

When the experiment involves real money, identity, credentials, financial access, transaction history and recovery, the warnings must be direct. Serious risks should not be softened merely because honest disclosure might slow adoption.

Until its dependencies, permission controls, recovery process and wallet-separation capabilities have been independently demonstrated, BSV Browser should be treated as an **experimental operating wallet**—not as the primary home for substantial savings, irreplaceable assets or sensitive identity credentials.

Scope and Evidence Discipline

This is a review of a changing product at a defined point in time.

The product findings in this paper are based on the BSV Browser repository, production-build configuration, support documentation, privacy policy and App Store release history available on August 6, 2026. The source snapshot was the `master` branch at commit `2d51bdb346156349d82f1a1b6c6166b3588b2d82`.

At that time, the Apple App Store identified version 1.5.5, released July 30, 2026, as the current iOS release, while the repository snapshot reported package version 1.5.6. This review does not assume that the distributed App Store or Play Store binaries were built from that commit.

That distinction matters.

The App Store independently confirms that version 1.4.5 introduced automatic approval of “small-amount spends.” The exact 100,000-satoshi threshold and ten-second cooldown come from the identified source snapshot. The report therefore distinguishes between:

- what the released application publicly announced;
- what the reviewed source code contained; and
- what would still require binary comparison or runtime testing to prove about the distributed build.

The public repository is actively changing. Important source files should therefore be preserved with screenshots, archived copies and commit-specific links as part of the publication evidence package.

The code or product may change after the review date. A future release may correct some of the problems identified here. That would be welcome.

The larger lessons are intended to last longer than any one software version.

Each important conclusion should be understood as one of five things:

Confirmed documentary fact
Something expressly shown in official source code, configuration or documentation.

Documented contradiction or omission
Two official statements that do not fit together, or an important matter the public documentation does not adequately explain.

Reasonable architectural inference
A risk that logically follows from confirmed design features without claiming that the feared conduct is occurring.

Unverified runtime question
Something that cannot be settled from documentation and source code alone.

Corrective recommendation
A safeguard that should be implemented or publicly considered, not a claim about the present product.

This is not a penetration test, packet-level network inspection, supply-chain audit or independent comparison of the store binaries against the public source.

Part One

What Ordinary Users Need to Understand

1. BSV Browser Is Becoming a Gateway

The easiest way to misunderstand BSV Browser is to think of it as a normal mobile browser that happens to hold a little BSV.

That is not what it is.

When Web3 mode is enabled, websites operating inside the browser can ask the wallet to create and sign transactions, provide public keys, encrypt and decrypt information, create signatures, manage identity certificates and perform other BRC-100 operations.

The wallet also keeps transaction and output information, manages trusted certifiers and communicates with services used to route messages, process transactions, follow the chain and support wallet activity.

The convenience comes from putting everything together.

The danger also comes from putting everything together.

Think about the modern smartphone. It is useful because the camera, microphone, contacts, location, payments, identity and applications are all in one place.

But that concentration also makes the smartphone one of the greatest collection and control points ever created.

Most people do not understand every setting. They do not know what is running in the background. They cannot personally verify what information leaves the device. They rely on defaults, permission screens and promises made by the companies involved.

BSV Browser creates the same kind of problem for BSV.

A person may technically control the private keys while having little practical understanding of:

- which outside services the wallet contacts;
- what those services can observe;
- which permissions remain active;
- when payments can occur automatically;
- how identity information may become connected to activity;
- which wallet records must survive for complete recovery;

- and whether the person can leave the original wallet without losing important information or access.

This is the difference between **cryptographic control** and **practical control**.

Holding the private key is essential.

It does not automatically answer every other question.

2. The Wallet Sets the Minimum Level of Safety

BRC-100 separates two actors.

The application asks.

The wallet authorizes and executes.

An application may ask for one payment, one certificate field, one signature or access to one basket. The wallet decides how the application is identified, what the user is shown, whether permission has already been granted, whether the request is approved and which supporting infrastructure is used.

That creates a basic rule:

> **An application can limit the risks it creates, but it cannot replace protections the user's wallet does not provide.**

A responsible application can ask for only what it needs. It can avoid unnecessary identity certificates. It can refuse to request key-linkage information. It can limit its own records. It can clearly explain why it is asking for money or information.

Those protections matter.

But the application generally cannot repair a wallet that:

- does not securely establish which website is making the request;
- remembers permissions too broadly;
- automatically approves repeated spending;
- mixes identified and pseudonymous activity in one database;
- relies on a service the user cannot replace;
- or cannot be completely recovered outside the original implementation.

Every interaction through BSV Browser therefore inherits the BSV Browser risks relevant to that interaction.

This does not mean every application inherits every possible problem.

An application that never requests a certificate does not create a certificate-disclosure risk of its own. But it still cannot fix weak wallet-side origin authentication or make the wallet's recovery records independently portable.

The application asks.

The wallet remains the gate.

3. Replacing Gatekeepers—or Moving the Gate?

BRC-100 is often presented as a way to stop every website from creating another account, collecting another copy of the user's information and trapping the user inside another private system.

That is a worthy goal.

Project Babbage publicly says centralized systems of control are inherently prone to abuse. It says users should control their own identities and data, move between platforms and make informed choices about how their information is used. Ty Everett is identified as the founding president of Project Babbage.

Those are the right principles.

But principles do not enforce themselves.

The BSV Browser source reviewed sets its default MessageBox service to:

`https://messagebox.babbage.systems`

MessageBox is used for peer-to-peer communications, including identity-based payments. The wallet-settings source contains visible controls allowing users to select or enter an ARC transaction-processing provider. No corresponding ordinary-user MessageBox-provider control was identified in the settings source or public support guide reviewed for this paper.

Put that in normal language:

> **Project Babbage says users should not be trapped behind one company or one gatekeeper. But when BSV Browser handles identity-based payment messages, it points them by default to Project Babbage's own MessageBox service—without plainly telling ordinary users or giving them an obvious setting to choose someone else.**

That does not prove that Project Babbage is reading messages, building user profiles or abusing the service.

It proves something more modest but still important:

The system currently places a Babbage-branded service in the middle of a feature marketed as direct identity-based payment, and the user-facing explanation is not good enough.

A message relay is not inherently improper. When a recipient's phone is offline, something may need to hold and deliver a message when the person reconnects. BRC-33 describes that relay role.

The problem is not simply that a relay exists.

The questions are:

- Why is this particular provider selected?

- What can it see?
- What does it keep?
- Who legally operates it?
- Can the user choose another provider?
- Can the user self-host?
- Can different providers communicate?
- What happens if the service is unavailable, pressured, sold or changed?

Even when the message contents are encrypted, a relay may still see useful surrounding information such as connection timing, request frequency, network addresses or message size. Exactly what the present MessageBox service can see, log or reconstruct has not been publicly explained.

BRC-34 already describes a federated approach in which different relay providers can cooperate, removing the need for everyone to depend upon one central server. The specification also says there were no known implementations of that federation at the time its page was reviewed.

This is not mainly about embarrassing one developer or company.

It is a warning about how gatekeepers are actually born.

They are not always created by villains announcing plans to control everyone. They are often created by people solving an immediate problem, choosing the easiest default and telling themselves that the service is benign, temporary or replaceable in theory.

Then applications are built around it.

Users accumulate history around it.

Other developers assume it will always be there.

The temporary default becomes the practical standard.

By the time anyone admits that a gate has been built, the ecosystem may already depend on the gatekeeper.

That is why the present MessageBox arrangement matters.

It gives the BSV community an early chance to correct the direction before convenience hardens into dependence.

4. Disclosure Is Not Public Relations. It Is a Security Control.

The BSV Browser support guide makes broad and reassuring promises.

It says:

> “Every wallet action requires explicit approval.”

It also says:

> “No payment is ever made without your explicit approval.”

And in its frequently asked questions:

> “You must explicitly tap Authorize before any funds leave your wallet.”

The current source snapshot does something different.

It defines a default automatic-approval threshold of 100,000 satoshis and a ten-second global cooldown between auto-approved transactions. The settings screen labels the user control “Auto Spend Up To” and allows the threshold to be changed or switched off.

That is a documented contradiction.

The support guide also says keys and transaction data are stored only on the device. The README similarly describes no WAB and local storage as the hardcoded defaults.

But the public production-build profile contains configured BSV Association WAB and storage endpoints, along with Babbage MessageBox, ARC and Chaintracks endpoints. The README simultaneously says production environment variables override local values.

The current wallet-initialization source may still override the hosted WAB and storage settings with no WAB and local storage. The production configuration may be outdated, inactive or reserved.

That cannot be settled from the documents alone.

What can be said is this:

> The public record contains two different stories about the production configuration, and ordinary users should not have to read source code to find out which one controls their financial wallet.

A configured endpoint does not prove that information is transmitted to it, logged by it or retained.

It does prove that the endpoint exists in the published production profile and requires an explanation.

The privacy policy says BSV Association does not collect or store certain personal information, while also acknowledging that third-party technologies and services may be integrated.

That may accurately describe what BSV Association itself retains.

It does not tell users what each wallet service receives.

A meaningful disclosure should say, service by service:

- what the service does;
- who operates it;
- which information is sent;
- whether network addresses or identity information are visible;

- whether information is logged;
- how long records are kept;
- which country’s laws apply;
- and whether the user can switch providers.

Saying “we do not hold your private keys” is not enough when another service may receive transaction or identity-related information.

Saying “everything stays on your device” is not enough when the application necessarily communicates with network services.

Saying “every payment requires approval” is not enough when the released application publicly advertises automatic small-payment approval and the source contains an auto-approval threshold.

An honest gatekeeper tells users what the gate does.

5. Recovery: Owning the Key Is Not the Same as Recovering the Wallet

The recovery problem may be the easiest issue for ordinary users to understand.

A person sees twelve recovery words and naturally assumes:

> If I lose my phone, those words bring my wallet back.

The BSV Browser support guide initially reinforces that belief. It calls the phrase the only way to recover the wallet and says it can restore the wallet on a compatible device.

Later, the same guide explains something much more serious.

The recovery phrase restores the root key only. The guide says there is no chain-scanning or derivation-path process capable of rebuilding the transaction history, UTXOs or certificates from the blockchain. Complete recovery requires a separate exported database.

That is not a minor technical footnote.

It changes what “recovery” means.

The user must protect two different things:

1. The secret capable of recreating the controlling keys.
2. The changing database that tells the wallet what has happened and what it currently controls.

The exported file is SQLite-based, not a secret or inherently proprietary database format.

The concern is that the complete wallet state appears to depend upon an implementation-specific database whose successful recovery into a separately developed wallet has not been publicly demonstrated.

That creates a form of **informational custody**.

The wallet company may never possess the private key needed to sign a transaction. Yet the wallet's records may still be essential for locating, organizing, understanding or restoring the user's assets and credentials.

The user may hold the key while remaining dependent upon the original wallet's records and recovery process.

BSV Browser does provide an important escape route for ordinary BSV. Its Legacy Bridge lets users send funds to conventional P2PKH addresses, including through a "Send Max" option. That is a real positive safeguard and should be preserved.

But moving ordinary BSV to an address does not prove that everything else can move cleanly.

The unresolved questions include:

- Can another independently developed wallet restore the exported database?
- Can certificates be recovered without BSV Browser?
- Can token and basket information be moved?
- Can application-linked outputs be identified and used elsewhere?
- Can the user leave without preserving a lifetime dependence upon one database schema?
- What happens if the original wallet disappears before the most recent database export?

A wallet is not truly replaceable merely because another developer could theoretically implement BRC-100.

The user must be able to leave.

6. Automatic Spending: The Human Wallet Is Becoming an Agent Wallet

The public support guide promises that every spending request requires the user to tap Authorize.

The App Store release history says version 1.4.5 added automatic approval of small payments.

The public source snapshot reviewed defines:

- a default threshold of 100,000 satoshis; and
- a minimum ten-second interval between automatic approvals.

The settings screen allows the user to change the threshold or turn automatic spending off.

At the BSV market price on the review date, 100,000 satoshis—0.001 BSV—was worth only about one cent.

The concern is not that one automatically approved payment would empty a wallet.

The concern is that:

- the documentation says every payment requires approval;
- the product publicly announced an exception;
- the source sets a default exception;
- no cumulative hourly, daily or monthly limits were demonstrated;
- no per-application total budget was demonstrated;
- and no protection against dividing a larger charge into repeated smaller charges was demonstrated.

The value of a fixed satoshi threshold may also change materially if BSV’s price rises.

This matters today as a wallet-security issue.

It may matter far more tomorrow as an AI-agent issue.

AI agents cannot operate efficiently if a human must manually approve every micropayment, data request and service call. Some form of delegated spending may be necessary for serious agentic use.

But delegated spending does not require giving an agent access to the human owner’s main wallet.

That would be the dangerous shortcut.

BRC Opinion 151 warned that an interface built for autonomous software may create disproportionate risks when it becomes the gateway to human savings, identity and personal information.

The proper answer is separation.

> **An AI agent should have its own wallet.**

That wallet should have:

- separate keys;
- a separate identity;
- a limited funded balance;
- no access to the owner’s reserve wallet;
- no automatic replenishment from savings;
- hard per-transaction limits;
- hourly, daily and monthly limits;
- restricted applications and protocols;
- expiring permissions;
- no authority to reveal the human’s general identity or key linkages;

- and an emergency stop the agent cannot override.

The limits must be enforced by the wallet or another system outside the agent’s control.

A natural-language instruction such as “do not spend more than $50” is not a financial control. An AI system can misunderstand instructions, be manipulated by hostile content or pursue a goal in an unexpected way.

The wall must be real.

7. Identity, Metadata and the Myth of a Single Safe Wallet

BRC-100 does not automatically publish a user’s identity key on-chain. Knowing the identity public key does not, by itself, allow an outside observer to derive every transaction key. Key-linkage revelation is not automatically public.

Those are genuine privacy protections.

The BRC-100 Identity Firewall Addendum nevertheless explains why the wider separation between identity and financial activity is not automatic.

Applications may request the identity key or certificates. Wallet and application records may preserve origins, transaction participation and other useful metadata. Authorized, compromised or legally compelled parties may later combine those records.

The danger is not that every identity payment instantly reveals the user’s entire financial history to the world.

The danger is that one wallet may hold enough separate pieces of information to make later correlation much easier.

BRC-52 identity certificates are designed to support selective revelation and UTXO-based revocation. A certificate is not the user’s private key, and spending a certificate’s revocation output indicates that the certificate has been revoked—not that the user’s ordinary BSV has been frozen.

One bad certifier does not automatically expose or compromise the entire wallet.

The gatekeeping risk appears when important applications begin demanding particular certificates, relying on a narrow group of approved certifiers or treating identified participation as the default.

A person may technically still own the wallet while becoming unable to participate in the useful application ecosystem without the accepted credential.

That is why certifier defaults, trust scores, update procedures and removal rights must be plainly disclosed.

The same principle applies to regulated assets.

A regulated token or financial service may legitimately require identity verification, recordkeeping, freezing, redemption or other legal controls.

But placing that identified activity inside the same wallet, database, provider account and recovery package as activity intended to remain pseudonymous creates unnecessary risk.

BRC Opinion 152 makes the distinction clear. Different baskets inside one wallet may help applications organize outputs or limit casual access, but the baskets may still share the same keys, history, database, storage, permissions and backup.

A folder is not a firewall.

A profile is not necessarily a firewall.

A basket is not a firewall.

The BSV ecosystem now needs four clear wallet roles:

Reserve Wallet

Long-term savings and high-value ordinary BSV, with no routine browser or application access.

Operating Wallet

Limited funds for normal human application use.

Agent Wallet

Limited funds, separate identity and hard controls for AI agents or other autonomous software.

Regulated Wallet

Identity-linked assets, regulated tokens and compliance credentials isolated from pseudonymous activity.

Each should be capable of using separate:

- key roots;
- identity keys;
- databases;
- permissions;
- provider settings;
- backups;
- and recovery procedures.

Separate risks with real wallets—not folders, baskets or promises.

8. Interim Guidance for Users

BSV Browser may be useful for experimentation, ordinary browsing, low-value application interaction and learning how BRC-100 systems work.

Users should nevertheless treat it as an actively changing operating wallet, not a proven vault.

Until the unresolved issues have been independently tested, users should:

- Keep substantial long-term savings in a separate wallet that is not continuously connected to applications.
- Set automatic spending to zero unless they knowingly accept the risk.
- Maintain both the recovery phrase and current wallet-database exports.
- Test recovery before relying upon it.
- Use the Legacy Bridge to verify that ordinary BSV can be moved to a conventional wallet.
- Avoid placing irreplaceable certificates or assets in the wallet unless independent recovery is understood.
- Keep regulated or identity-linked assets separate from activity intended to remain pseudonymous.
- Never allow an AI agent to operate through the primary human wallet.

The warning should not be hidden in a technical appendix:

> **Until its complete recovery, service dependencies, permission controls and wallet separation have been independently demonstrated, BSV Browser should be treated as an experimental operating wallet—not as the primary repository for substantial savings, irreplaceable assets or sensitive identity credentials.**

Part Two

Questions and Corrective Actions for Developers

9. Questions That Need Clear Answers

These questions are not accusations.

They are questions ordinary users should not need a source-code expert to ask.

Services and Data

1. Which outside services does the released product contact during browsing, payments, identity discovery, transaction monitoring and recovery?
2. Who legally operates each service?
3. What information can each service see?
4. What information is logged, and for how long?
5. Is `messagebox.babbage.systems` the only ordinary-user MessageBox option?
6. When will users be able to choose, self-host or federate MessageBox providers?
7. Are the hosted WAB and storage endpoints in the production-build configuration active, obsolete or reserved?

Permissions and Spending

8. How is the domain displayed in a permission screen securely bound to the actual WebView page, frame and application session that made the request?
9. Which wallet permissions persist after approval?
10. Where can users review and revoke every permission previously granted?
11. Why does the support guide say every payment requires approval when the released application announced automatic small-payment approval?
12. What cumulative limits prevent repeated automatic charges or payment splitting?
13. Is automatic spending prominently disclosed during wallet creation?

Recovery and Exit

14. Can the complete wallet be recovered using an independently developed wallet?
15. Which assets and records require the BSV Browser database?
16. Can certificates, baskets, tokens and application-linked assets be exported in a documented standard format?
17. Has recovery been independently tested after loss of the original device and disappearance of the original wallet provider?

Identity and Separation

18. Which certifiers are trusted by default, including those inherited from wallet libraries?
19. Who can change the default certifier list?
20. Can users maintain multiple simultaneously active wallets with completely separate keys, identities, databases, providers, permissions and backups?

21. Can regulated assets be automatically placed into a genuinely separate wallet?
22. Can AI-agent wallets be created with hard limits enforced outside the agent?

Verification

23. Do the App Store and Play Store binaries reproducibly match the public source?
24. Has an independent security, privacy, recovery, dependency and software-supply-chain audit been performed?
25. Which recommendations in BRC-151 and BRC-152 have been accepted, rejected or scheduled?

10. Corrective Action Program

A. Tell Users What Is Actually Happening

BSV Association should publish a short, readable **Wallet Exposure Statement**.

It should not be buried in source code or written only for developers.

It should identify:

- every outside service used by the released application;
- the company or organization operating it;
- the purpose of the connection;
- the information sent and received;
- logging and retention practices;
- whether the user can replace the provider;
- every form of automatic authorization;
- what the recovery phrase restores;
- what the database restores;
- what can be moved to another wallet;
- and which risks remain experimental or untested.

BRC-151 already recommends separate application-side and wallet-side disclosures.

Applications should explain what they ask for.

Wallets should explain what they permit, remember, store, reveal and require for recovery.

Any material change should produce a visible notice and renewed consent.

A privacy statement should not say everything stays on the device while leaving users to discover network services in the source.

A support guide should not promise approval for every payment while the product automatically approves some payments.

Correct the words first.

Then correct the code.

B. Give Users a Real Permission Dashboard

Users need one place to see:

- every application that has interacted with the wallet;
- every permission granted;
- every permission still active;
- every certificate field revealed;
- every basket accessed;
- every identity or key-linkage request;
- every automatic payment;
- and every spending limit.

From that screen, users should be able to revoke an application immediately without losing access to their funds.

Permissions should expire unless knowingly renewed.

Expanded access should always require fresh consent.

Spending controls should include:

- per-transaction limits;
- per-application limits;
- hourly limits;
- daily limits;
- monthly limits;
- protection against splitting one charge into many smaller charges;
- optional notifications;
- and a master emergency stop.

Automatic spending should be off by default for human wallets.

C. Let Users Choose the Services Standing Between Them and Bitcoin

A service that can be replaced only by editing source code and publishing a new application is not meaningfully replaceable for an ordinary user.

MessageBox, ARC, storage, wallet authentication, Chaintracks, identity discovery and certifier manifests should be visible in the wallet.

Where technically practical, users should be able to:

- choose among providers;
- enter a custom provider;
- self-host;
- use community-operated services;
- use multiple providers;
- switch without losing wallet state;
- and receive a clear warning when one provider becomes unavailable.

BSV Browser has already added visible ARC endpoint choice to the wallet settings.

That is a good step. It proves that provider choice can be presented in a way ordinary users can understand.

The same principle should apply to MessageBox.

BRC-34 already describes a path toward federated message relays. That path should be implemented, or the public should be told why it has not been.

D. Build Real Wallet Walls

BSV Browser should support multiple independent wallet instances as a first-class feature.

Each instance should be capable of having:

- a separate master key;
- a separate identity key;
- a separate database;
- separate permissions;
- separate provider settings;
- separate recovery material;
- and no automatic visibility into another wallet.

At minimum, the interface should encourage:

Reserve Wallet: savings, no general application access.

Operating Wallet: limited funds for everyday human use.

Agent Wallet: limited funds and hard controls for autonomous software.

Regulated Wallet: identity-linked assets and credentials isolated from pseudonymous activity.

The wallet should never describe a basket or visual profile as equivalent to this separation.

The user should know whether a wall is real.

E. Make Recovery Independent

A wallet must be able to fail without taking the user’s digital life with it.

The recovery system should be explained in plain language from the beginning:

- The recovery phrase restores the key.
- The database restores additional wallet state.
- Both may presently be required.
- Losing one may have different consequences from losing the other.

Developers should create and document an open recovery package that independent wallets can import.

That package should include everything necessary to restore:

- ordinary BSV;
- derivation and transaction records;
- spendable outputs;
- certificates;
- baskets;
- token information;
- application-linked assets;
- and necessary provenance or overlay information.

Independent developers should be invited to demonstrate successful recovery without BSV Browser.

Recovery should be tested publicly through a simple challenge:

> Delete the application, remove the original vendor from the process and restore everything into a separate implementation.

Until that can be done, claims of vendor neutrality remain incomplete.

F. Audit the Released Product, Not Merely the Repository

Public source code is valuable.

It does not prove that the store binary matches the repository, that network behavior matches the documentation or that the permission system behaves safely under attack.

An independent review should include:

- reproducible-build comparison;
- network-traffic inspection;
- origin-authentication and permission testing;
- automatic-spending abuse and payment-splitting tests;
- recovery testing;
- corrupted-database testing;
- provider-outage testing;
- certifier-manifest testing;
- cross-wallet portability testing;
- AI-agent containment testing;
- third-party dependency review;
- software-supply-chain and build-pipeline auditing;
- update-signing and release-control review;
- and testing of the actual distributed binaries.

The findings should be public.

Serious unresolved issues should be disclosed in the application before users create a wallet.

Financial infrastructure should not rely on quiet production testing using people’s real money.

11. Minimum Implementation Checklist

Before BSV Browser—or any future browser-wallet gateway—is promoted as a primary user wallet, developers should be able to answer **yes** to the following:

- Are all outside services plainly disclosed?
- Can users choose or replace routing and transaction providers?
- Are application origins securely authenticated?
- Can users review and revoke every permission?
- Is automatic spending off by default for human wallets?
- Are cumulative and per-application spending limits enforced?
- Can the complete wallet be recovered without the original vendor?

- Can ordinary BSV leave through a conventional address?
- Can certificates, tokens and wallet state move to another implementation?
- Are reserve, operating, agent and regulated wallets truly separate?
- Are certifier defaults and governance public?
- Do released binaries reproducibly match the public source?
- Has the software supply chain and build pipeline been independently reviewed?
- Has the product passed an independent security, privacy and recovery audit?
- Are experimental limitations prominently disclosed before users place money or identity into the wallet?

A “no” answer does not necessarily mean the product must stop existing.

It means the risk should be stated plainly and the missing safeguard should have a public correction plan.

Conclusion

BSV Browser has the potential to become something valuable.

A browser that allows users to carry money, identity, permissions and credentials between applications could reduce today’s dependence upon countless accounts, passwords and private databases.

But there is another possible future.

Instead of eliminating gatekeepers, the system could move the gate into one dominant wallet.

Instead of every website holding a piece of the user, one wallet and its selected services could sit between the user’s money, identity, applications and recovery.

Instead of freeing people from platforms they cannot leave, the ecosystem could create a wallet they cannot completely recover without.

That outcome does not require bad intent.

It only requires convenience to keep winning over separation, disclosure and user choice.

The present BSV Browser implementation gives the community an early warning.

The MessageBox default shows how one company’s service can quietly become part of the path between users.

The automatic-spending contradiction shows how quickly a human-approval promise can give way to software convenience.

The recovery process shows how a person can hold the keys while still depending upon the wallet’s records.

The lack of demonstrated wallet separation shows how savings, agents, identity and regulated assets could slowly collect behind one common gate.

These problems can still be corrected.

Project Babbage and BSV Association have publicly embraced user control, privacy, open standards and freedom from centralized intermediaries.

The answer is not to abandon those principles.

The answer is to apply them to their own products.

Tell users what the wallet is doing.

Let them choose the services it uses.

Give them real walls between different kinds of risk.

Make recovery work without the original provider.

Prove that the released product matches the promises.

And do it before users place years of money, identity, credentials and history behind a gateway they cannot easily leave.

The central lesson is larger than BSV Browser:

> **A system does not empower the user merely because the user holds the private key.**

The user must also be able to understand the system, limit it, separate it, recover it and leave it.

Appendix A

Plain-English Glossary

These definitions explain how important terms are used in this paper. They are written for ordinary readers rather than software developers.

ARC
A service used to send a signed transaction to the BSV network and track what happens afterward. ARC may report whether the transaction was accepted, rejected or included in a block. It does not hold the user’s private keys.

Automatic spending / auto-approval threshold
A wallet setting that allows certain payments to be approved without showing the user a separate confirmation screen each time.

Basket
A way for a BRC-100 wallet to label and group particular transaction outputs. A basket can help organize assets or limit application access, but it is not the same as a completely separate wallet with different keys, records, permissions and recovery material.

BRC
Short for **Bitcoin Request for Comments**. BRC documents describe technical standards, proposed designs and accepted opinion papers used in the BSV ecosystem.

BRC-100
A common interface through which an application can ask a compatible wallet to perform operations such as creating transactions, providing public keys, signing or encrypting information, managing certificates and listing wallet records.

BRC Opinion 151 / BRC-151
The accepted opinion paper titled *BRC-100 Risk Assessment and Best Integration Practices*. It examines centralization, surveillance, security, recovery and ecosystem-control risks that may arise when one wallet interface combines money, identity, applications and broad software authority.

BRC Opinion 152 / BRC-152
The accepted opinion paper titled *Best Practices for Regulated Tokens in a BRC-100 Ecosystem*. It explains why regulated or identity-linked assets should be kept in a genuinely separate wallet rather than merely placed in a different basket inside the same wallet.

Certificate / identity certificate
A digitally signed claim about a person or organization. A certificate might confirm age, name, membership, nationality or another attribute. A certificate is not the user’s private key, and revoking it does not automatically revoke ordinary BSV funds.

Certifier
A person, company or organization that issues an identity certificate and vouches for the information stated in it.

Chaintracks
A service or system used to follow BSV block headers and help a wallet check whether transaction proofs connect to the accepted blockchain.

Conventional Bitcoin address / P2PKH address
The familiar address-based method of sending BSV. P2PKH means **Pay to Public Key Hash**.

Federation
An arrangement in which several independent service providers communicate instead of forcing every user through one central provider.

Identity key
A persistent cryptographic public key used by a wallet to represent the user in identity and authentication interactions. It does not automatically reveal a legal name but may become connected to real-world identity through certificates or other activity.

Informational custody
A situation in which a company or wallet may not hold the private key needed to spend the user's money but still controls or supplies important information needed to locate, understand or fully recover the wallet and its assets.

Legacy Bridge
The part of BSV Browser that lets users send or receive ordinary address-based BSV payments.

MessageBox
A relay service that temporarily receives and delivers messages between users, especially when the recipient is offline. In BSV Browser, it supports identity-based peer-to-peer payment communications.

Originator
The application or website identified as making a request to the wallet.

Permission
Authority given to an application to ask the wallet to perform an operation involving spending, certificates, encryption, signatures, baskets or other wallet functions.

Presentation key
A 256-bit key managed through a Wallet Authentication Backend and used for authentication or authorization. It is distinct from the ordinary private keys that spend the user's BSV.

Reference implementation
A working example created to show developers how a standard can be implemented. Its design choices may become influential because others copy, reuse or build around them.

Regulated asset
A token, payment product or financial asset subject to identity verification, recordkeeping, freezing, redemption, compliance or other legal controls.

Remote storage
Wallet records kept on a server or system outside the user's device.

Recovery phrase / seed phrase
A sequence of words used to recreate a wallet's controlling root key. It may restore the keys without necessarily reconstructing every record kept by a particular wallet.

Runtime testing
Testing the actual installed application while it is operating, including its permission screens, recovery behavior and network connections.

Self-custodial wallet
A wallet in which the user controls the private keys needed to spend the funds. Self-custody does not automatically eliminate dependence on outside routing, storage, authentication or recovery services.

Trust manifest
Information published by a certifier or trust provider describing its identity, services and trust information.

UTXO
Short for **Unspent Transaction Output**. A UTXO is an individual spendable piece of BSV created by an earlier transaction.

Wallet Authentication Backend / WAB
An optional outside authentication server. A WAB may use methods such as SMS or identity verification to create or retrieve a presentation key used to authenticate a returning user. It does not necessarily hold the ordinary private keys that spend BSV, but it may become part of the user's login or recovery path.

Wallet database
The records a wallet keeps about transactions, UTXOs, certificates, baskets and other wallet state. BSV Browser documents an SQLite-based `.db` export that preserves information the recovery phrase alone does not reconstruct.

Wallet separation
Using genuinely different wallets for different risks. Real separation normally means separate keys, identity, database, permissions, backups and provider settings—not merely different labels, baskets or screens.

Appendix B

Primary Sources Reviewed

Review Date and Source Snapshot

This paper reviews BSV Browser and its public supporting materials as publicly available on August 6, 2026. The source-code snapshot reviewed was the `master` branch at commit `2d51bdb346156349d82f1a1b6c6166b3588b2d82`.

BSV Browser is actively changing. Product findings describe the source code, documentation, configuration and released-application information available at that time.

Important source-code pages should be preserved as screenshots, PDFs and commit-specific links with the publication archive. A moving `master`-branch link may show different code after the repository is updated.

A. Controlling BRC Opinion Papers

BRC Opinion 151

BRC-100 Risk Assessment and Best Integration Practices
Centralization, Surveillance, Security and Ecosystem-Control Risks

Official accepted BRC Opinion:
https://bsv.brc.dev/opinions/0151

BRC Opinion 152

Best Practices for Regulated Tokens in a BRC-100 Ecosystem
Preventing Identity Spillover and Cross-Activity Correlation

Official accepted BRC Opinion:
https://bsv.brc.dev/opinions/0152

BRC-100 Risk Assessment Addendum

The Bitcoin White Paper's Identity Firewall and BRC-100

BitcoinSV.Guide research paper supplied for this review.

B. BRC Standards and Technical Specifications

BRC-100 — Wallet-to-Application Interface

https://bsv.brc.dev/wallet/0100

BRC-33 — PeerServ Message Relay Interface

https://bsv.brc.dev/peer-to-peer/0033

BRC-34 — PeerServ Host Interconnect Protocol

https://bsv.brc.dev/peer-to-peer/0034

BRC-42 — Key Derivation Scheme

https://bsv.brc.dev/peer-to-peer/0042

BRC-46 — Output Baskets

https://bsv.brc.dev/wallet/0046

BRC-52 — Identity Certificates

https://bsv.brc.dev/peer-to-peer/0052

BRC-69 — Key-Linkage Revelation

https://bsv.brc.dev/wallet/0069

BRC-116 — Wallet Permissions and Counterparty Trust

https://bsv.brc.dev/wallet/0116

C. BSV Browser Product and Source Materials

Main Public Repository

https://github.com/bsv-blockchain/bsv-browser

Production-Build Configuration

`eas.json`

https://github.com/bsv-blockchain/bsv-browser/blob/master/eas.json

Reviewed for publicly configured WAB, storage, MessageBox, ARC and Chaintracks endpoints.

Default Configuration

`context/config.tsx`

https://github.com/bsv-blockchain/bsv-browser/blob/master/context/config.tsx

Reviewed for local-storage, no-WAB and MessageBox defaults.

Wallet Initialization and Permission Logic

`context/WalletContext.tsx`

https://github.com/bsv-blockchain/bsv-browser/blob/master/context/WalletContext.tsx

Reviewed for wallet initialization, permissions, automatic spending, MessageBox configuration and certifier settings.

Spending Constants

`shared/constants.ts`

https://github.com/bsv-blockchain/bsv-browser/blob/master/shared/constants.ts

The reviewed source lines define:

- `DEFAULT_AUTO_APPROVE_THRESHOLD = 100_000`
- `AUTO_APPROVE_COOLDOWN_MS = 10_000`

Wallet Settings

`app/wallet-config.tsx`

https://github.com/bsv-blockchain/bsv-browser/blob/master/app/wallet-config.tsx

Reviewed for user-facing automatic-spending, ARC, trust, recovery and database controls.

Wallet Database Import

`utils/importDatabases.ts`

https://github.com/bsv-blockchain/bsv-browser/blob/master/utils/importDatabases.ts

Reviewed for BSV Browser's SQLite database-import process.

Support and User Guide

https://mobile.bsvb.tech/support.html

Reviewed for wallet permissions, explicit payment approval, local storage, recovery, certificates, trusted certifiers and Legacy Bridge.

Privacy Policy

https://mobile.bsvb.tech/privacy.html

Reviewed for claims concerning personal information, browsing records, local data and third-party services.

Apple App Store Listing and Release History

https://apps.apple.com/us/app/bsv-browser/id6755637932

Version 1.5.5, released July 30, 2026, was listed as the current iOS release at the time of review. The version 1.4.5 release notes stated that small-amount spending was automatically allowed.

D. Supporting Infrastructure

Wallet Authentication Backend

https://github.com/bsv-blockchain/wab

The standalone WAB repository was archived in June 2026 after development moved elsewhere.

ARC

https://bsvblockchain.org/features/arc/

Chaintracks

https://github.com/bsv-blockchain/go-chaintracks

E. Public Accountability and Gatekeeper Statements

These sources are included to compare publicly stated anti-gatekeeper principles with the actual service dependencies found in BSV Browser. They are not offered as proof of malicious intent or improper data use.

About Project Babbage

https://projectbabbage.com/about

Project Babbage X Posts

July 22, 2026 — Main BRC-100 manifesto

https://x.com/ProjectBabbage/status/2080078479010492525

July 22, 2026 — Competing implementations and switching

https://x.com/ProjectBabbage/status/2080066121416511979

https://x.com/ProjectBabbage/status/2080068400483856649

July 31, 2026 — Privacy statement

https://x.com/ProjectBabbage/status/2083264139393393053

August 25, 2024 — Surveillance-capitalism statement

https://x.com/ProjectBabbage/status/1827503183104765984

Screenshots, print-to-PDF copies and full-thread context should be preserved separately because social-media posts may later be deleted, hidden or made difficult to retrieve.

F. Research and Reconstruction Materials

The following supplied documents were used as research, comparison or drafting material:

1. *BRC-100 Risk Assessment and Best Integration Practices — BRC Opinion 151*
2. *BRC-100 Risk Report Addendum 1*
3. *Best Practices for Regulated Tokens in a BRC-100 Ecosystem — BRC Opinion 152*
4. *BSV Browser Preliminary Risk Review (T7 Grok)*
5. *BSV Browser Preliminary Risk Review — Deeper Technical Analysis (T7 Gemini)*
6. *BSV Browser: Risk Review, Unanswered Questions and Corrective Recommendations — Prior Draft (T7 Claude)*

The preliminary reports and prior draft were treated as research material, not as independent proof. Important technical conclusions and public statements were rechecked against the primary sources listed above. Final Report was prepared by T7 ChatGPT; with final edits, fact checks and suggestions by T7 Kimi, T7 Claude, and T7 Gemini.
