KNOWLEDGE 💡 FOUNDATION

Knowledge Foundation : https://KnowledgeFound.org | Knowledge Token® : https://KnowledgeFound.org/token

COLLABORATORS

- Immersive Education Initiative (USA)
- Lucerne University of Applied Sciences and Arts (Switzerland)
- Massachusetts Institute of Technology (MIT) Bitcoin Club (USA)
- University of Oxford Blockchain Research Centre (UK)
- University of Zurich Blockchain Center (Switzerland)
- Yale University Blockchain Club (USA)
- Brown University Blockchain Club (USA)

INTERNATIONAL PARTNERS

- United Nations Global Resource for Anti-Corruption Education and Youth Empowerment (GRACE) initiative

- United Nations Office on Drugs and Crime (UNODC)

2024 SOUTH AFRICA
BLOCKCHAIN HACKATHON

Stellenbosch University | February 06 - 07

The inaugural **South Africa Blockchain in Education Symposium** and hackathon will be hosted by **Stellenbosch University** on February 6th and 7th in collaboration with **South Africa's National Institute for Theoretical and Computational Sciences** and the Switzerland-based **Knowledge Foundation** international standards organization.

Table of Contents

1. AWARDS / PRIZES

Join us at Stellenbosch University or online this February 6th and 7th to compete in South Africa's inaugural blockchain hackathon for a chance to win R1,279,876 ($70,000) in awards.

Developer grant prizes are awarded by the DFINITY Foundation in collaboration with the Knowledge Foundation. The grants are used to fund participation (after the hackathon) in developing Knowledge Token® on the Internet Computer platform and will be distributed after the hackathon has ended.

The Knowledge Token® prizes will be awarded to the winning teams and solo participants and are redeemable towards registration fees at all future Summits and Symposia worldwide.

Award (prize) details for the 2024 South Africa Blockchain Hackathon are located on the **Hackathon** page of the official event website.

2. REQUIREMENTS

2.1 OPEN SOURCE

All submissions (entries) for this hackathon will be made available to the public in a free and open manner, and are therefore automatically considered Open Source upon submission. Only Open Source submissions will be considered.

2.2 SUBMITTING YOUR SOLUTION (ENTRY)

At the conclusion of the hackathon you will submit your solution (hackathon entry) using an online form that will be provided to you during the hackathon, at which time you will be required to provide:

1. A link to your source code. For this you can use a public GitHub repository, a public website folder, or any other public online code repository.

2. A link to your solution running on the Internet Computer main network ("mainnet"). In order to do this you must deploy your solution to an Internet Computer canister on the mainnet and "fuel" your canister using "cycles" (see "FREE CYCLES" below).

3. A link to your design documents, if you created such materials. Note that design documents are not required, but are recommended.

4. The name and email address of each member of your team (or yourself, if you are competing Solo as an individual and not on a team).

During the hackathon you will receive the official submission form so that you can provide the above information.

2.3 BLOCKCHAIN PLATFORM

Only hackathon entries (submissions) created/programmed using the open source Internet Computer will be considered.

Internet Computer is a relatively new blockchain platform for which you will find the following resources important to become familiar with this platform in time to compete.

A link to the DFINITY Foundation, the non-profit foundation in Switzerland that is responsible for Internet Computer, is also provide (DFINITY Foundation has the largest team of researchers and developers in the field of blockchain, and are our hackathon collaborators):

1. DFINITY Foundation: https://dfinity.org
2. INTERNET COMPUTER: https://internetcomputer.org
3. IC DASHBOARD: https://dashboard.internetcomputer.org

See DEVELOPER RESOURCES below for Internet Computer developer documentation, videos and support resources.

2.4 PROGRAMMING LANGUAGES

Your hackathon entry (submission) may be programmed in any language supported by the Internet Computer. Please note, however, that for this particular hackathon the Rust language may be particularly helpful as Rust libraries exist for the Bitcoin Integration feature of Internet Computer that is a critical aspect of this hackathon.

Below you'll find examples of how to utilize Bitcoin Integration using the Motoko language as well.

Note that Motoko is preferred but not required (see JUDGING PREFERENCES below).

2.5 BITCOIN INTEGRATION

The hackathon has a special focus on the Bitcoin Integration feature of Internet Computer. Only entries (submissions) that use the Bitcoin Integration feature of Internet Computer will be eligible for this hackathon:

- **Bitcoin Integration overview**
  https://internetcomputer.org/how-it-works/bitcoin-integration

- **Bitcoin Integration with Motoko**
  https://internetcomputer.org/docs/current/tutorials/developer-journey/level-4/4.3-ckbtc-and-bitcoin

Use the Bitcoin test network ("testnet"). Do not use the Bitcoin main network ("mainnet"). Bitcoin mainnet requires real Bitcoin; use the Bitcoin testnet instead.

2.6 FREE CYCLES

At the conclusion of the hackathon you will submit a link to your program running on the Internet Computer main network (mainnet). To do this you must deploy your solution to the Internet Computer mainnet and fuel your canister with cycles.

You can obtain free cycles via the **Cycles Faucet** detailed in the official Internet Computer developer documentation below:

- **Developer Documentation**:
  https://internetcomputer.org/docs/current/developer-docs (see "Acquiring and using cycles" and "Mainnet Deployment")

- **Video**: https://www.youtube.com/watch?v=eynEk3Bz7QY&t=902s

- **Developer Journey**:
  https://internetcomputer.org/docs/current/tutorials/developer-journey

- **Developer Journey Discussion forum**:
  https://forum.dfinity.org/t/developer-journey-feedback-and-discussion/23893

2.7 DEVELOPER RESOURCES

DEVELOPER DOCUMENTATION:

- **Internet Computer Developer Documentation**
  https://internetcomputer.org/docs/current/developer-docs

DEVELOPER DISCUSSION FORUM AND DISCORD:

- **Internet Computer Developer Discussion Forum**
  https://forum.dfinity.org

- **Internet Computer Developer Discord**
  https://discord.com/invite/jnjVVQaE2C

DEVELOPER JOURNEY:

- **Developer Journey**
  https://internetcomputer.org/docs/current/tutorials/developer-journey

- **Developer Journey Video Tutorial Series (YouTube videos)**
  https://www.youtube.com/watch?v=oBUpJ4CqmN0&list=PLuhDt1vhGcrdR2h6nPNylXKS4u8L-efvD

- **Developer Journey Discussion Forum**
  https://forum.dfinity.org/t/developer-journey-feedback-and-discussion/23893

Judging Preferences

The following recommendations are not requirements, but are important "judging preferences" that are applied during the judging phase of the hackathon.

Solutions (entries) that are otherwise equivalent will be awarded higher scores when the following preferences are followed:

- **MOTOKO Preference**: Preference is given to solutions that code back-end functionality in Motoko, the open-source programming language created specifically for the Internet Computer.

- **SVELTE Preference**: Preference is given to solutions that code front-end functionality using Svelte, the open-source front-end component framework.

- **SINGLE-PAGE Preference**: Preference is given to single-page solutions where all user interactions occur on the same web page (as opposed to interfaces that use multiple web pages), or using a minimal number of web pages when single-page solutions are not possible.

- **MOBILE-FIRST Preference**: Preference is given to "mobile-first" solutions. Mobile devices (phones, tablets, etc.) are the primary target, although your solution's user interface should still be functional using desktop/laptop computers.

- **DESIGN DOCUMENTS Preference**: Preference is given to solutions that are accompanied with corresponding design documents (architecture designs, data designs, flow diagrams, user interface mock-ups, interaction diagrams, etc.)

4. HACKATHON TARGETS (OBJECTIVES)

Targets (Objectives)

The objective of this hackathon is to construct a **Bitcoin Donation System** that anyone with a web browser can use to donate any amount of Bitcoin to specific aspects of the Knowledge Token® open International Standards project.

Note that you will use the Bitcoin test network ("testnet"), not the Bitcoin main network ("mainnet"). Bitcoin mainnet requires real Bitcoin. Use the Bitcoin testnet, for which free test Bitcoin is available.

The Bitcoin Donation System consists of two "targets" that together comprise the hackathon:

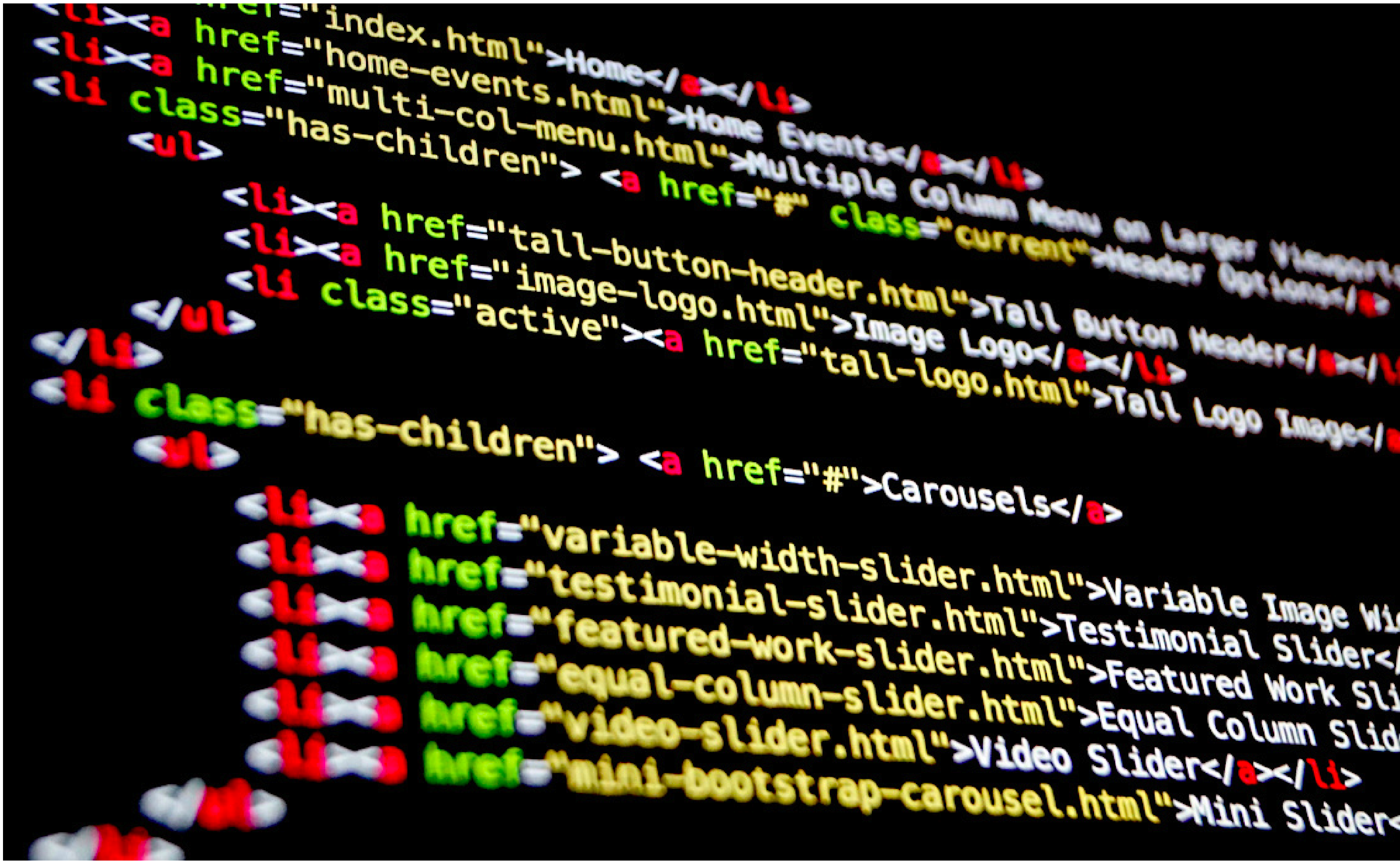
- **Front-end user interface** (see FRONT-END TARGET below)
- **Back-end logic and functionality** (see BACK-END TARGET below)

Hackathon competitors will design and develop (code) both a front-end and back-end to produce a working solution that is deployed and running on the Internet Computer main network (mainnet), with Bitcoin functionality running on the Bitcoin testnet.

Competitors will submit a link to their working solution for judging at the end of the hackathon (see REQUIREMENTS above and SUBMITTING YOUR SOLUTION below).

Note that all hackathon participants will create both a front-end and back-end for their solution (competition entry). Only working solutions that have both a front-end and back-end will be accepted.

5. FRONT-END TARGETS



```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li><a href="multi-col-menu.html">Multiple Column Menu on Larger Viewports
<li class="has-children"> <a href="#" class="current">Header Options</a>
  <ul>
    <li><a href="tall-button-header.html">Tall Button Header</a></l
    <li><a href="image-logo.html">Image Logo</a></li>
    <li class="active"><a href="tall-logo.html">Tall Logo Image</a
  </ul>
</li>
<li class="has-children"> <a href="#">Carousels</a>
  <ul>
    <li><a href="variable-width-slider.html">Variable Image Wi
    <li><a href="testimonial-slider.html">Testimonial Slider</
    <li><a href="featured-work-slider.html">Featured Work Sli
    <li><a href="equal-column-slider.html">Equal Column Slid
    <li><a href="video-slider.html">Video Slider</a></li>
    <li><a href="mini-bootstrap-carousel.html">Mini Slider<
  </ul>
</li>
```

Please note that the following instructions are the full extent of detail provided to hackathon competitors. The instructions intentionally do not provide technical specifications, nor technical guidelines. A primary goal of the hackathon is to enable competitors to design their own solutions based on a minimal set of general guidelines.

Acceptable Technologies

Your front-end (user interface) solution may be developed using any combination of traditional Web technologies that are recognized or endorsed by the World Wide Web Consortium (W3). This includes HTML, XML, Cascading Style Sheets (CSS), JavaScript, ECMAScript, and JSON.

3rd party platforms, code libraries, extensions, additions, and so forth may be utilized provided no Intellectual Property Rights (IPR) encumbrances, restrictions or licenses apply that might prevent or limit the use or distribution of your hackathon code to the global public in a free and open manner.

Note that mobile-first solutions are preferred, but not required. See JUDGING PREFERENCES above for details.

Note that Svelte is preferred, but not required. See JUDGING PREFERENCES above for details.

Front-end Target 1 : Bitcoin Donation User Interface (BDUI)

Your first target for the front-end is a **Bitcoin Donation User Interface (BDUI)** that enables anyone with a web browser to donate any amount of Bitcoin to schools and/or students. Your solution should take a mobile-first design approach (designed primarily for use on mobile devices, such as phones and tablets), yet still be accessible and functional using desktop/laptop computers.

Your solution should provide a simple, clean and easy to understand interface that allows end users to select a school, or a student at a school, to donate Bitcoin to. Your user interface must further allow the user to specify exactly how their donation should be allocated across (divided between) the various donation categories as noted below (see DONATION CATEGORIES below).

Note that a single-page user interface is preferred for the Bitcoin Donation User Interface (BDUI), but not required. See JUDGING PREFERENCES above for details.

Users of your solution should be able to:

- + Choose a school, or a student at a school, to make a donation to
- + See one or more photo(s) of each school and the students at each school
- + Specify how much Bitcoin in total to donate to a specific school or student
- + Specify how much of their total Bitcoin donation should be allocated (subdivided) across the "donation categories" noted below
- + Look up the detailed record of their donation using a transaction identifier that is unique to their donation (see **Front-end Target 2** below)

SELECTING SCHOOLS AND/OR STUDENTS

Your solution should present the end user with the ability to select a school, or a specific student at a school, that they would like to donate Bitcoin to.

- Your solution should provide several schools (and several students at each school) for the user to choose from.
- The user should be presented with images (photos of schools and students) and text descriptions (school names and locations, student names and grade levels, etc.) from which they can choose one. The one (school or student) they select is the one that will receive the donation.

PRIVACY NOTICE:

Do not use real (actual) student photos and names unless they are of you and your team. Instead, use public photos (such as Creative Commons photos or "stock photos"), or generate fictitious student photos yourself using generative AI technology such as Stable Diffusion, Midjourney, etc.

Be careful to also create fictitious names for students in your solution.

SPECIFYING TOTAL AMOUNT OF BITCOIN TO DONATE

Your solution should allow the end user to specify exactly how much Bitcoin in **total** they would like to donate to the school or student they select. The amount should be represented as a decimal number with 8 decimal places (ex., 1.00000000) since one Bitcoin is divisible to eight decimal places.

SPECIFYING CATEGORY AMOUNTS (SUBDIVIDING THE DONATION)

Your solution should allow the end user to specify what amount of their total donation should be allocated to each of the donation categories (see below). This amount can be specified as a percentage of the total or an exact amount of Bitcoin.

DONATION CATEGORIES

End users must allocate their total donation across the following categories:

- Curriculum design and development
- Teacher support
- School supplies
- Lunch and snacks

EXAMPLES

Your user interface must allow the user to specify how to allocate their total Bitcoin donation across the categories either using exact Bitcoin amounts (a specific amount of Bitcoin per category) or using percentages of the total amount being donated.

In the examples below exact Bitcoin amounts are used for demonstration purposes.

For example, if a user donates a total of 4.00000000 Bitcoin they might split that amount evenly across all categories by allocating 1.00000000 Bitcoin to each category as illustrated below:

```
TOTAL: 4.00000000

Curriculum design and development:     1.00000000
Teacher support:                       1.00000000
School supplies:                       1.00000000
Lunch and snacks:                      1.00000000
```

Or, for example, they may wish to allocate the total amount to a single category as illustrated below (in this example all four Bitcoin are donated for "School supplies"):

```
TOTAL: 4.00000000

Curriculum design and development:     0.00000000
Teacher support:                       0.00000000
School supplies:                       4.00000000
Lunch and snacks:                      0.00000000
```

Likewise, the user can split the total donation between the categories in any way they choose. For example:

```
TOTAL: 4.00000000

Curriculum design and development:     0.50000000
Teacher support:                       1.00000000
School supplies:                       2.30000000
Lunch and snacks:                      0.20000000
```

By default donations are split evenly (equally) across the four categories, which the end user can change as they see fit. Your solution should therefore pre-set each category with an equal share based on the total donation amount.

Note that your solution should give the end user (donor) a way to provide you with their Bitcoin transaction ID upon successfully sending Bitcoin to your donation wallet. Their Bitcoin transaction ID should be part of their donation record. See FRONT-END TARGET 2 below and BACK END TARGET below.

Note that the total amount of Bitcoin is sent to a single master wallet, and that all donations from all donors are sent to this single master wallet. See BACK-END TARGET below for details.

Front-end Target 2 : Donation Transaction Explorer (DTE)

After an end user has made a donation they should be provided with a unique **Donation Transaction Identifier (DTI)** that they can use at any time to look up the details of their donation.

In addition, anyone with a web browser should be able to view the details for every donation that has been made. All donation record details are public.

Your solution should therefore provide a **Donation Transaction Explorer (DTE)** that allows open access to all donation records and also the ability to "look up" a specific donation record by its Donation Transaction Identifier.

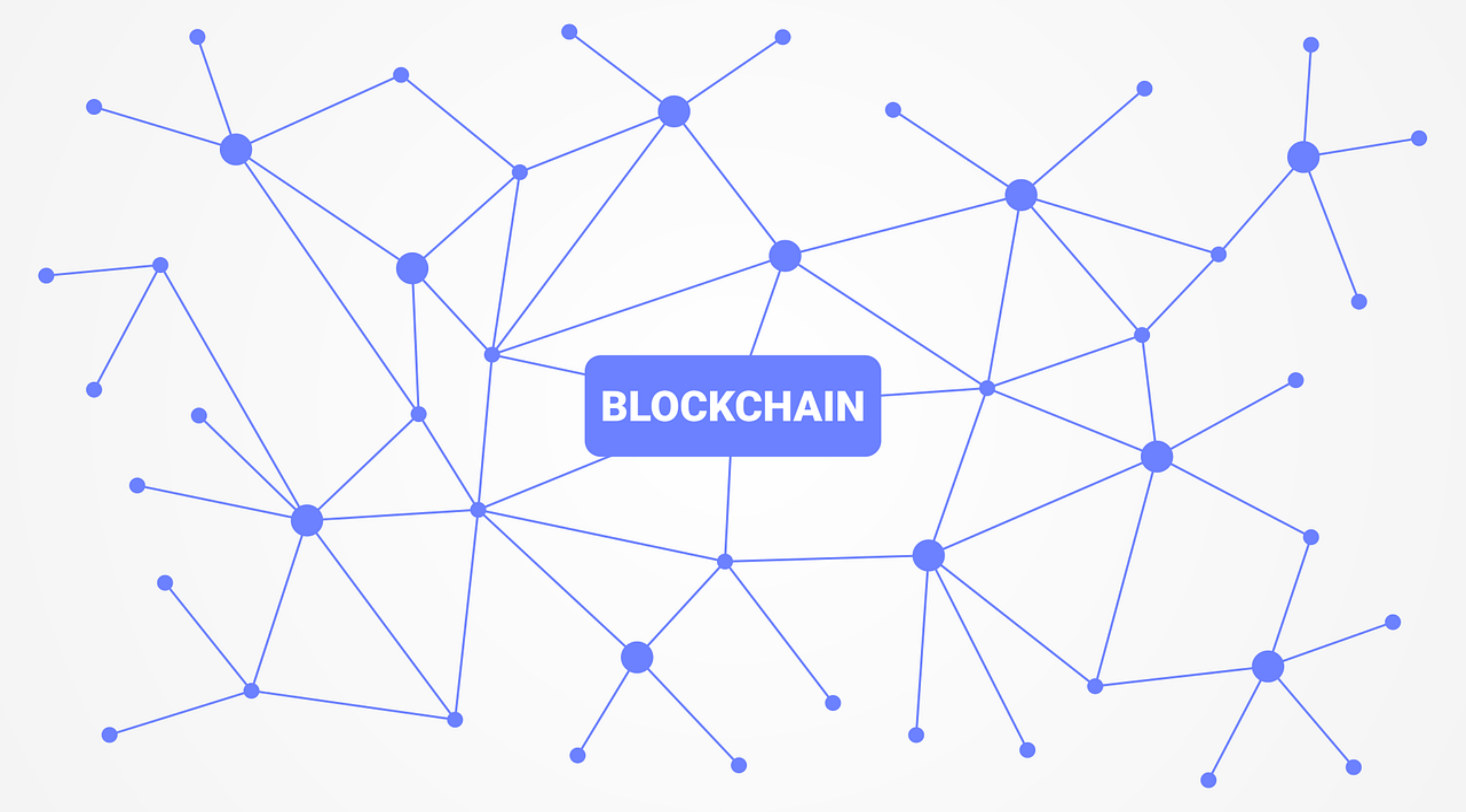
Donation transaction records ensure that:

1. Bitcoin donations are allocated according to the wishes of donors,
2. Donors can review the details of their donations at any time, and
3. Anyone can examine the full collection of donation records

*Note that the Donation Transaction Identifier (DTI) is **not** the same as the Bitcoin transaction ID that end users (donors) will furnish to you upon successfully sending Bitcoin to your donation wallet. See BACK END TARGET below for details.*

Note that a single-page user interface is preferred for the Donation Transaction Explorer (DTE), but not required. See JUDGING PREFERENCES above for details.

6. BACK-END TARGET



Please note that the following instructions are the full extent of detail provided to hackathon competitors. The instructions intentionally do not provide technical specifications, nor technical guidelines. A primary goal of the hackathon is to enable competitors to design their own solutions based on a minimal set of general guidelines.

Acceptable Technologies

Your back-end solution must be created using the Internet Computer and Bitcoin Integration specifically to enable the following front-end targets:

- **Front-end Target 1 :** Bitcoin Donation User Interface (BDUI)
- **Front-end Target 2 :** Donation Transaction Explorer (DTE)

BITCOIN DONATION WALLET

Note that the total amount of Bitcoin that a user donates is sent to a single wallet. Do not create a wallet for each donation category, and don't create a donation wallet for each user.

Simply create a single Bitcoin donation wallet that receives all donations from all users and record (via donation transaction records) the details for each and every donation that is made. The transaction details that you record for each donation provide a comprehensive account of all donations made, including details about how funds are to be allocated across the donation categories.

PUBLIC DONATIONS

Anyone with a web browser should be able to make a donation: an Internet Computer account is not required. To facilitate public donations simply provide donors with the address of your solution's donation wallet so that donors can send Bitcoin funds to your donation wallet, and also provide a way for the donor to furnish you with their corresponding **Bitcoin transaction ID (TXID)**.

The Bitcoin transaction ID that the end user provides to you should become part of their donation record. A Donation Transaction Identifier (DTI) is used to uniquely identify a donation record, and is **not** the same as the Bitcoin transaction ID. See FRONT-END TARGETS above for details.

It is suggested that you link the Bitcoin transaction ID to a corresponding public Bitcoin blockchain explorer record (such as provided by https://blockstream.info) in order to provide a way for anyone exploring your solution's donation records to see the underlying Bitcoin transaction details of each donation.

The Bitcoin transaction ID (TXID) that donors provide to you is the transaction ID (transaction hash) of the Bitcoin "send" transaction they will conduct from their own wallet to send funds to your donation wallet. The donor's Bitcoin wallet will provide a TXID to them upon sending funds to your donation wallet. TXID is not the same as the **Donation Transaction Identifier (DTI)** described above.

DONATION RECORDS

How you record the details of each donation is up to you. You can use any of the available data structures provided by the Internet Computer, for example, or you might consider creating a custom JSON object and store those in a data structure or file on the Internet Computer.

Whatever mechanism you use must run entirely on the Internet Computer.

Note that your entire solution, including how you represent, store and retrieve donation records, must run on the Internet Computer. It is not permissible to use services that run outside of the Internet Computer.

7. ONLINE AND IN-PERSON PARTICIPATION

ONLINE

Hackathon participants that compete online should join the hackathon Zoom meeting on February 6th at 9:00 AM local time (Time Zone: Stellenbosch, South Africa).

The Zoom meeting will remain open throughout February 6 and 7. Online competitors that reside far outside of the time zone should join as soon as possible if they can't join at 9:00 AM local time (you will still be able to compete even if you can't attend the initial online Zoom meeting).

For instructions on how to join the Zoom meeting **RELOAD** the following hackathon page at 9:00 AM local time on February 6:

https://summit.ImmersiveEducation.org/SouthAfrica/2024/hackathon-start.html

IN PERSON

Hackathon participants that compete in person at Stellenbosch University should be in the hackathon room on February 6th at 9:00 AM local time (Time Zone: Stellenbosch, South Africa).

In-person competitors should also be prepared to join the Zoom meeting while on campus in the hackathon room at Stellenbosch University.

For instructions on how to locate the hackathon room at Stellenbosch University (and also Zoom meeting instructions) **RELOAD** the following hackathon page the day before the hackathon begins on February 5:

https://summit.ImmersiveEducation.org/SouthAfrica/2024/hackathon-start.html

8. SUBMITTING YOUR SOLUTION

SUBMITTING YOUR SOLUTION

All hackathon entries must be submitted by February 7th at 3:00 PM (15:00) local time (Time Zone: Stellenbosch, South Africa)

For instructions on how to submit your entry **RELOAD** the following hackathon page at 3:00 PM (15:00) local time on February 7:

https://summit.ImmersiveEducation.org/SouthAfrica/2024/hackathon-start.html