Replication Guidelines
Open source solutions for Public Service Delivery

Evdokia Kouraki, Johanneberg Science Park
Camilla Sandberg, Sebastiaan van Herk, Bax & Company
Table of Contents

1. Introduction .................................................................................................................... 3

2. Background and definitions .................................................................................... 5
   What is replication? ........................................................................................................ 5
   What are open source solutions for public service delivery? ....................................... 5
   How to improve service delivery through open source .............................................. 6

3. Understanding the replication process of open source smart city solutions ............... 8
   The replication process ............................................................................................... 8
   Barriers for replication ............................................................................................. 11
   Data as a conversation starter .................................................................................. 12
   Factors for replicability ............................................................................................ 13

4. How to replicate open source smart city solutions step by step ................................ 18

5. Concluding remarks ................................................................................................. 28

References.......................................................................................................................... 29
This guideline for replication of open source solutions for Public Service Delivery (PSD) is created within the scope of the Interreg NSR project SCORE (Smart Cities Open data Re-use). The aim of SCORE is to increase efficiency and quality of public services in cities and communities based on replicable smart and open data-driven solutions. The objective of the guidelines is to help project managers, coordinators and decision makers engaged in smart city projects to facilitate the replication of such open source smart city solutions from one city to another. They highlight technical aspects of replication, essential considerations for assuring an enabling context for successful replication, as well as how this links to overall initiatives of digital transformation. The guidelines can also be useful for developers and programmers who want to know more about the context for preparation and implementation to be able to work with their city on replication of open solutions.

At the time of the publication of these guidelines, the solutions under development within the project consist of a combination of open source and non-open source code. Efforts are made to increasingly provide more code open source. This guide is written prior to completing the replication processes for the SCORE solutions which means that the final lessons from this process are not incorporated. Instead, the guide makes use of existing material on replication of smart cities solutions and adds a specific focus on open source software in order to provide guidance to the SCORE partners for how to facilitate and prepare for replication. The content in this document has been developed based on information from interviews with representatives from partners engaged in the SCORE project and other related smart city projects, as well as from reports and deliverables created by other European projects and organisations working on the topic.

The step-by-step guide links the replication activities to other deliverables within the SCORE project such as the guide for testing in Urban Living Labs previously developed by Johanneberg Science Park, the tools and outcomes of the Peer Review process led by Bradford university and the preparation of Business cases for the solutions facilitated by the City of Ghent.

The SCORE project initiated from the ambition to improve public service delivery and take city challenges as a starting point for solution development. Importantly, SCORE promotes and develops a collaborative and open approach to build shared solutions that address common city challenges. The work of SCORE is thus much in line with, and endorses the recent initiatives for a ‘European way’ for the digital transformation of sustainable cities and communities supported by the European Commission and the Committee of the Regions1. This replication guide is a resource for the EU’s digital transformation in that it directly guides on some of its core principles: (i) deploying innovative digital solutions tailored to individual needs, (ii) development of efficient, cost-effective and citizen-centric services, (iii) scaling up of open, interoperable, cross-sector and cross-border platforms.

1 https://www.living-in.eu/
The guidelines cover three main topics:

Definitions and background information
Defining the main concepts and explaining the rationale behind SCORE.

Understanding the replication process of open source smart city solutions
Explaining the process, barriers, drivers and factors of replication.

How to replicate open source smart city solutions step by step
Providing details on steps to take, questions to ask and decisions to make in a replication process.

The SCORE Project - Open Source solutions for Public Service Delivery

The SCORE partners leverage open collaboration methodologies – open source, open data, open standards and an open attitude – to learn together, communicate effectively, share challenges and collaborate on solutions with the goal of improving public service delivery in our cities.

SCORE (Smart Cities Open data Re-use) is a collaborative project between nine cities throughout the North Sea Region aimed at enhancing public service delivery based on smart, data-driven solutions. The cities in the SCORE project co-define shared challenges and thereafter pool resources and expertise to (co-)develop at least 12 innovative solutions. The solutions are categorised in four areas: ‘mobility’, ‘water’, ‘environment’, and ‘transversal’. Replicability is at the center of the project. Reusing already developed code from another city allows for cities to take already tested solutions and/or components and build on them to meet their needs. The aim of the project is that the replication of solutions and reuse of code will lead to a reduction of the time needed for development of solutions. Overall, the project aims to increase the efficiency and quality of public service delivery and to reduce the costs for delivering these services. With the ambition to develop and provide the solutions free and open source, SCORE contributes to a more transparent and inclusive way of offering efficient public services.

SCORE Partners cities and their participating organisations:
Amsterdam (Amsterdam City and Amsterdam Data Science University), Aberdeen City, Aarhus (Aarhus City and Aarhus University), Bergen City, Bradford (Bradford City and University of Bradford), Dordrecht City, Ghent (Ghent City and Digipolis Ghent), Gothenburg City (Johanneberg Science Park), and Free and Hanseatic City of Hamburg.

Project Period: 2018-2022
Total budget: €5.855.000
Interreg NSR/ERDF contribution: €2.670.000
Project Website: https://northsearegion.eu/score/
SCORE pages on GitHub: https://github.com/score-partners
Replication should be understood as a wider concept of reusing different aspects and components of the developed solutions.

The principles of open source promote collaboration and co-creation.

Open source software can improve public service delivery, reduce costs and assure transparency and sovereignty.

**What is replication?**

The word replication refers to reproducing something in exactly the same way. However, replication of smart digital solutions for public service delivery is usually more complex than taking a solution developed to solve a specific challenge in one city and copying it to another city. With the objective of improving public service delivery and reducing cost and time for development, a broader definition of replication is more valuable. Reflecting this, the Smart Cities Information System (SCIS) knowledge platform has defined replicability in the following way:

> Replicability refers to the possibility of transporting or ‘copying’ results from a pilot case to other geographical areas, albeit with potentially different boundary conditions. In other words, if a pilot was proven to work in one community or region, it could be exported to other communities or regions (indigenously or abroad), but taking into account that the boundary conditions could be quite different from those in the piloted community or region. Replication may also encompass the management process that was used in the pilot scheme or the cooperation structure between critical stakeholders.

In this sense, the understanding of replication in SCORE and in this guide is that replication can mean copying a full solution, however, it is more likely to refer to reusing parts of a solution by taking generic components either directly, or adapting or repurposing them to function in a different context. The perspective of replication of management and cooperation processes, as defined by SCIS, is also recognised and referred to as contextual factors in this guide (see chapter 3 for more explanations). Transnational replication between different cities is the objective of SCORE. However, replication may also happen within cities, where new use cases are found for the same technical components.

**What are open source solutions for public service delivery?**

Open source solutions for public service delivery refer to the smart, data-driven solutions to specific challenges in cities based on software developed according to principles of open source. Types of public services that can be facilitated through these smart, data-driven solutions include for instance better management of sustainable mobility, improving air quality, monitoring flooding and furthering crowd management. Open source means that the source code of the software is accessible for anyone to inspect, modify, enhance and share (in modified or unmodified form). Typically, open source software is made available with distribution terms that complies with the criteria by the Open Source Initiative (OSI) and the Open source definition.

The principles of open source promote collaboration and co-creation. The network of developers, which open source development potentially taps into, stimulates innovation.

---

2. **Background and definitions**


    3. [https://opensource.org/osd](https://opensource.org/osd)
More people working together on the same solution helps to ensure that solutions can meet a broader range of needs, that mistakes are quickly exposed and corrected, and that the software is continuously enhanced.

**How to improve service delivery through open source**

Public authorities are increasingly exploring the potential of open source software in their own developments as well as for procured products. Open source software can be a way for cities to offer high quality public services, reduce costs and increase control. In comparison to procurement of traditional software, where there is a high dependency on the vendor, their licensing agreement and their interest in sharing data and making customisations, open source software can provide more flexibility. Adopting open source software is a way to maintain a level of independence and self-determination over the city’s technological infrastructure, in order to avoid proprietary (vendor) “lock-in”.

Open source software development can help to improve internal capacity for application development but it does not rule out procurement of development services. IT vendors are increasingly subscribing to the idea of open source and expand their business models around support for open source development for their clients. Furthermore, in line with the philosophies of open government, there is also increasing legislation that suggests the use of open data and open source, such as the EU Open data directive. Allowing for collaborative and open access to data as well as software to make use of it help to ensure technological sovereignty for cities and offer transparency towards citizens.

### 5 reasons for cities to work with open source software

1. Avoid vendor lockin (independence and self determination)
2. Knowing the code makes it easier to make corrections, customisations and adaptations
3. Avoid licensing fees
4. Cost savings from replication and reuse of code
5. Development of internal capacity for application development

As with any new strategy, there are of course limitations and barriers associated with using open source solutions for public service delivery. A main barrier for adopting open source software in public organisations is that it requires things to be done differently and change can be difficult. The digital architecture may need to be redesigned and organisational processes, including procurement practices and long-term maintenance of the software, need to be rethought and adapted. The collaboration principles of open source require the public authority to consider how to position itself in the open source community which can be complex for an organisation with no previous experience of this. Furthermore, open source is frequently perceived as more complex than traditional software development, as reusable software needs to be more generic, more multi-purpose and more adaptable.

---


---

6 Identified in the interviews with SCORE partners, found in the arguments for public code by Foundation for Public Code (https://standard.publiccode.net/introduction.html) and supported by Cassell, 2008 (ibid).
3. Understanding the replication process of open source smart city solutions

→ Important to understand how different generic technical components can be used for multiple use cases

→ The similarities of data and interoperability of systems can be a starting point for replication

→ A solution developer can facilitate replication by assuring findability, documentation and demonstrations, quality code and accessible repository, and an appropriate open source licence

The replication process

Most of the time, replication of a full solution is not possible, instead the solutions built for one city context must be appreciated as smaller components of functionalities (or in some cases actively broken down in components) before being adapted for use in a different context to create a new solution.

Figure 1: Overview of replication flow
The figure above aims to show different scenarios of replication from one city to another (could also be between different policy areas of the same city). The challenges that cities face are in many cases similar between different cities. These ‘shared urban challenges’ are then experienced locally and embedded in the unique context in each city. A solution, in SCORE defined as a digital application or library, can serve different functional purposes in a particular context in a city, so called ‘use cases’. Each solution consists of a number of ‘technical components’ creating different functionality. The technical components can have a more or less generic character making them more or less suitable to replicate for different use cases. To develop, test and implement a solution in a city, much work is needed in terms of engaging stakeholders, finding financing models or assuring that all aspects of a solution are working within the legal framework. The different aspects of assuring the context are here defined as contextual factors for which different processes need to take place to enable the adoption of the aspired new smart city solution. On a city level, there are approaches to these contextual factors that may serve as best practice examples for (re) creating an enabling context.

By identifying all these aspects (‘components’) of a smart city solution, we can find several variations of replication - replication of a full solution, replication of certain technical components but for different use cases or replication of use cases and functionality but with different code. It is unlikely to find cases where a solution can be replicated as an exact copy to a different context. So instead of looking at replication of a full solution, it is recommended to appreciate the solution as different components (or actively break it down into different components) and then see how some of these components can be used for different use cases in your city, contributing to a new solution to the challenges of your citizens and your public service delivery. Many SCORE partners testified that while the full replication scenario is the most attractive, the component scenario has the most potential to succeed.

Figure 2: Different aspects of a solution

<table>
<thead>
<tr>
<th>Technical components</th>
</tr>
</thead>
<tbody>
<tr>
<td>The different technical components that are developed to make the solution work for the intended use cases. The components can be of more or less generic character.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>The concrete context to which the solution is or can be applied in a city.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Contextual factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual factors and the different approaches to these developed by cities to create an enabling context for the implementation of the solution. For instance: stakeholder engagement, financing model, legislative framework, political support, user acceptance, digital structure etc.</td>
</tr>
</tbody>
</table>
The QR Toolkit as a SCORE solution emerged from an interest to experiment using user-friendly, low level prototypes to test IoT solutions as well as to find an affordable alternative to sensors that can be used on a large number of objects in a city. Attaching a unique QR code per item, any mobile device functions as a bridge to IoT-like features, such as geolocation, accessible citizen reporting and interactive contextualisation.

At the moment of writing, the QR (Quick Response) Code Toolkit developed in SCORE, is split in four components: QR Generator, QR Registration, QR Workflows, and QR Admin. This allows a potential open source reuse in other applications, mainly by customizing the last two components: the Workflows are responsible for what the end-user sees and depend on the object to which the QR code is assigned, and the Admin deals with the management of the different categories of objects and their settings. In this way every city department can personalize the use of the codes, adapting the QR-tagged objects in order to make them an integral part of a larger open information exchange system, potentially “communicating” among themselves.

The generic functionality of the technical components of the solution makes it possible to easily adapt to different use cases. Different use cases of the QR toolkit solution are currently being implemented in the SCORE cities Bradford and Ghent. In Bradford the solution has been applied to trash bins in the streets: each bin has a unique QR code that can be scanned by citizens and signalled if it is too full or dirty. This solves the problem of a generic reporting using the citizen’s GPS when they take photos, resulting in ambiguous information (e.g. bins close to each other bins). In Ghent, one of the use cases is the tracking of movable road signs allowing for The Roads & Bridges department to better keep track of its signs, and the citizens to directly report the location of displaced road signs. Another one is the possibility for various city departments to provide further information related to the reason for placing the movable sign (e.g. during events).

The long-term ambition is that the QR Code Toolkit, thanks to its open source availability and the shared best practices of the co-developer and follower cities, can be replicated in any European city.

The application’s source code is available as a Github repository, so its development can be followed closely and improvements can be suggested.

For more information see: https://northsearegion.eu/score/score-solutions/qr-code-toolkit/
Barriers for replication

Even though there are many arguments to replicate and reuse what is already available, it is not always easy to make it happen. As with any other development in a city, it requires work to prepare an enabling context. The EU knowledge platform Smart Cities Information System (SCIS) has identified six overarching reasons why replication is not happening at the scale desired. Working with the stakeholders to increase knowledge and awareness about alternative solutions can help to overcome these barriers. Once the “we are unique syndrome” is broken down into which aspects actually are unique and where some standard components can work to address a certain issue, the decision makers can be more receptive to the input from another city. To build confidence in replicating a solution, complete information about technical and non technical aspects of the original development, testing and implementation is essential. Later on in this guide we look at the type of information that will help assess the replicability and make a plan for replication.

The barriers presented below are not necessarily overcome by working open source, rather, an open source solution is likely to face the same challenges. However, open source can help to strengthen the argument for change of practices and trigger innovation, both in terms of products and services as well as in new ways of working together. Open source and replication can together play an important role for how cities can improve their services to the citizens while reducing cost and time spent.

Figure 3: Common barriers for replication and potential solutions

Source: Authors elaboration based on information in SCIS report “Why replication may not be happening.

The digitalisation of cities and communities is moving fast and access to and use of different types of data is crucial. Through various mechanisms cities collect, create and share data which combined serves as the foundation for many of the services offered in cities. In addition to being a driver for services, the availability and need for certain data can often be the starting point for replication of solutions. The data that cities obtain often has much more similarities between cities than the fragmented technology the cities use to obtain and analyse it. Once access to similar data in compatible formats is assured, you can more easily align the technology that will make use of it. A software that is replicated will typically need to decouple the original data sources in use, then find and plug in the data from the replicator’s own data sources. By aiming to create solutions that honour legacy systems and use a novel yet simple data infrastructure it is possible to access, share and re-use existing data from cities without overhauling entire city IT operations. To facilitate replication, the software developed should aim to enable the interoperability of existing data.

Interoperability of data

Regardless if a solution is open source or not, the interoperability of digital platforms is crucial in an increasingly digitised world. Interoperability refers to the possibility of sharing data between different systems and devices. This makes it possible for more actors to access and use data from systems. Interoperability of public services across borders and cross sectors is promoted through the ISA² Programme adopted by the European Parliament and the Council of European Union in 2015 following ISA which ran from 2010-2015. The ISA² Programme works with coordination of interoperability activities at EU level. Among other things, the ISA² Programme has put in place: a revised European Interoperability Framework (EIF); a revised European Interoperability Strategy (EIS); an Architecture, the European Interoperability Reference Architecture (EIRA); and a Cartography of solutions, the European Interoperability Cartography (EIC)⁸.

The European Interoperability Framework defines basic interoperability guidelines in the form of common principles, models and recommendations for the delivery of European public services in an interoperable manner. To advance the interoperability agenda further, in December 2019, the European Council launched the ‘Join, Boost, Sustain declaration’ to urge governments and local administrations to “agree on some principles and actions to better use all possibilities of digital technologies to improve the quality of life of European citizens⁹”.

A way to approach an increased interoperability is to create and rely on larger shared frameworks for the definition of the minimum technical requirements that are needed to make sure that different IoT solutions can be interfaced with local authorities’ digital systems. A joint approach to minimum technical requirements is crucial to have an ecosystem based on shared and shareable data, that speeds up solutions building, reduces costs and vendor lock-in, and improves citizens engagement and wellbeing. For this purpose the H2020 project SynchroniCity (2017 - 2019) has defined and validated the SynchroniCity framework based on OASC Minimal Interoperability Mechanisms (MIMs)¹⁰. The OASC MIMs are common technical mechanisms established to enable local authorities and technology providers to easily exchange digital products, services and data.

The interoperability of data and the compatibility of digital services is also promoted by Connecting Europe Facility Connecting Europe Facility - CEF Digital¹¹. CEF Digital is funding the Building Blocks, a set of generic

---

⁸ https://ec.europa.eu/isa2/isa2_en
¹⁰ https://synchronicity-iot.eu/
¹¹ https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL
and reusable Digital Service Infrastructures (DSI). Currently eight Building Blocks are offered: Big Data Test Infrastructure, Context Broker, eArchiving, eDelivery, eID, eInvoicing, eSignature and eTranslation. The Building blocks are endorsed by the European Commission and have the objective to make public digital services fully compatible with others on the market.

Factors for replicability

As seen above, matching a smart city solution to a viable use case is the first requirement for successful replication. If it is not based on a real need and if there is not enough political support it is unlikely that the implementation will be successful no matter how well a solution is tested, functioning and prepared for replication. Creating the right context in terms of legal and financial framework as well as ensuring the support from end users and policy makers is essential for any replicating city to ensure success.

The potential to replicate a solution depends on the interest and support from the developer of the original solution to share and facilitate the necessary information. The interest of a solution developer to facilitate the replication is often founded in the belief that open collaboration among a community of developers leads innovation and qualitative solutions. When the code is shared the liability for the code and the interest to make it work well is also shared. If you help others use your code, they will also rely on it. Once they rely on it, they will also inspect it carefully, finding and fixing bugs and security issues. They will also help invest in keeping it up to date, and make upgrades, and may even add new things. With more contributors, the positive impact from the community increases. In the long term, this means that the codebase is applied in a different context and people will use it in different ways. This over time makes the codebase stronger, as it is approached and configured for multiple angles. Additionally, with more cities adopting the same solution the familiarity with the software among companies and experts will increase, allowing for economies of scale and more access to people who have the capacity to support with the software.

Looking beyond the contextual challenges for replication and focusing on how to increase the replicability of components from concrete solutions there are a few things that solution developers can do to facilitate the process. Four aspects are particularly relevant to increase replicability: the solution and its components must be ‘findable’, it needs proper documentation and access to demonstrations, the code should meet high quality standards, and it should be accessible through an appropriate licence. When identifying if a solution developed somewhere else is suitable for replication, the information collected in the SCORE Peer Review process provides good insights.

Replicability factors

→ Findability
→ Documentation and demo
→ Quality Code and accessible repository
→ License

12 Important factors for replicability identified in interviews with SCORE partners.
1. Make sure people can find your solution

A first step of replication is to be aware of and understand what is already available. Most developers already have the habit of searching for how a similar solution has been programmed before. When it comes to city representatives, they also get informed about other solutions through their networks and existing projects. However, so far there is no single or de facto standard overview of solutions, like Github for open source developers, for cities to start from.

Within SCORE, the Challenge visualiser & explorer 13 set up by Ghent and Digipolis is a tool to visualise common challenges between the cities participating in the project. On a larger scale, the new OASC Catalogue, with the objective to build a global repository of certified implementable solutions by cities for cities, can support cities searching for new solutions. The catalogue will be covering IOT solutions and components related to different challenges and value propositions. Other platforms include for instance the collection of solutions on Joinup 14 and the CEF Digital Building Blocks 15. Tools like these can further enable cities and the different agencies of the cities to identify potential solutions to be replicated.

To increase findability of a solution it is recommended to:

- Chose a name for the solution that explains what it does.
- Consider a brief screen capture movie that explains the challenge, intended users and core functionalities.
- Consider a working demo (or the ability to install a quick demo).
- Consider a small project website and search engine optimisation.

13 https://northsearegion.eu/score/score-solutions/challenge-visualiser-explorer/
14 https://joinup.ec.europa.eu/
15 https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home
2. Provide clear documentation and easily accessible demo

All solutions need to have concise and full versions of documentation. For international replication the documentation needs to be in English and stored and organised in a way that is logical to anyone interested in the solution. Repositories should be stored in a suitable platform. In addition to written documentation, an easily available demo version makes it possible for a city representative to quickly see how the solution works and how it can be applied to the particular needs in the city. Demo versions and documentation is a way to assess the maturity of the solution and understand if the code is already widely adopted or if it is still early stage testing.

Documentation of the solution development project is recommended to include:

- Technical documentation stored on an open platform and available in English
- Link to demo version or visualisation of final solution
- Screenshots, which can be annotated to focus on specific core features
- Concise and complete description of non-technical aspects of the project (see template at the end of this document)
- Generic contact information and specific contact persons. Consider a direct contact for city representatives and one for developers
- Information on timing and status of the project
- Information on current project reuse, where, when and by who is it replicated or co-developed. What is the current status for each replication

3. Ensure quality code and accessible repository

To ensure that a solution or a component can be reused by others the source code and the information related to the code need to have a certain quality level. With co-creation of public service solutions and replication in mind, the SCORE partners have developed a set of code quality standards that guide developers and hold them accountable for the solutions that are developed\(^\text{16}\). The SCORE Principles for Quality Code specify what projects ‘must’, ‘should’ and ‘may’ have.

\(^\text{16}\) The guidelines are shared on Github, so for the latest version, always refer to the page: [https://github.com/score-partners/quality-code](https://github.com/score-partners/quality-code)
### Projects MUST
- have a README with a description of the codebases for the potential users/implemeters
- explain in the README how to run, develop and build
- be developed in a public repository from the start
- provide a link to a public list of known issues and bugs
- allow non-contributors to post issues, bugs and suggested changes
- have documentation for all public APIs in the [OpenAPI standard](https://openapi.io)
- have a LICENCE with an [Open Source Initiative designated “Popular License” or EUPL](https://opensource.org/licenses)

### Projects SHOULD
- kick ass
- have functional tests
- have a high level description for (technical) management to understand in the README
- respond to incoming issues and change suggestions within a week
- provide contributors guidelines in a CONTRIBUTING file
- set up any tests to run in a public continuous integration environment
- have test and documentation code coverage to the highest degree possible
- have documented private APIs
- use a version number compatible with [Semantic Versioning](https://semver.org)
- provide documentation on privacy and security
- contain configurations for automatic deployment and provisioning through systems such as OpenShift, Docker, Ansible, CloudForge etc
- Use the [GitFlow branching model](https://gitflow.readthedocs.io)

### Projects MAY
- publish a roadmap
- provide a forum or mailing list for discussion
- publish packages to relevant language specific repositories such as PyPi, Ruby Gems, NPM, Puppet Forge, etc
- provide a contribution and governance model in a GOVERNANCE file
- indicate the level of maintenance provided by whom and until when
- indicate a codebase/project status in the README that is one of: "Ideation"; "Alpha"; "Beta"; "Production"; "Archival"
- provide a list of contributors in a CONTRIBUTORS file

---

*Figure 4: SCORE principles for quality code*
4. Help others to use it - Select a suitable license

The licences of a software determine requirements and restrictions for how to use the software. Depending on the use case of a solution, different licenses are more or less suitable. The legal terms of the licence for open source software is different from proprietary software. In general, open source licenses grant the permissions to use the software in the way you want. The main issue when deciding on the type of license is the requirements and restrictions that the license places on users, there are two main types:

Copyleft (Strong: GPL, Microsoft Public/Weak: CDDL, MPL, Eclipse):
- Other people have the right to use, modify and share the work as long as the reciprocity of the obligation is maintained.
- Whoever uses such an open source license must make his/her code open for use as well.

Permissive (MIT, Apache, BSD):
- guarantees the freedom to use, modify and redistribute, while also permitting proprietary derivative works, placing minimal restrictions on how people can use open source components
- the code can be used freely in the way the user wants as long as you provide attribution back to the original developers and do not hold them liable.

The most commonly used open source licenses during 2019 (according to GitHub’s choosealicense.com) in order of popularity are: MIT, Apache-2.0, GPLv3, GPLv2, BSD 3, LGPLv2.1, BSD 2, Microsoft Public, Eclipse 1.0, BSD.
4. How to replicate open source smart city solutions step by step

→ For a replicator, there are five main steps to take for replication, in each step, there are a number of aspects to be assess and decisions to make.

→ A replication process is not necessarily completely linear and sequential, some activities may happen in parallel and several iterations might be needed to reach the final solution.

**Figure 5: Replication step by step**

<table>
<thead>
<tr>
<th>SCOPING</th>
<th>PREPARATION</th>
<th>ADAPTATION</th>
<th>TESTING &amp; IMPLEMENTING</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify need</td>
<td>2. Define solution</td>
<td>3. Ensure context</td>
<td>4. Make adaptations</td>
</tr>
<tr>
<td>5. Test and prepare to implement</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. **Identify need**

As for any other development in a city, the implementation of new data driven smart solutions is only useful when it corresponds to an actual need of the end user and contributes to improving public service delivery. The starting point for the implementation of new solutions is identifying the challenges that the city faces in general and how the citizens and city operators experience this. As the challenges are many, the problem needs to be given some priority to be addressed. If the challenge and the potential solution is in line with a larger political agenda and specific priority areas, it is more likely that an enabling context will be created. A demand driven development approach based on actual needs has a greater potential for actually improving the public service delivery.

The needs for a solution can be assessed on two levels, directly or indirectly. There can be a citizens’ need that can be directly met by giving them access to a certain service, however, often the smart city solutions help users from different departments or contracted operators in the city to improve the overall service delivery in a field.

Questions to consider:

- What are the challenges in the city?
- What do the citizens want/need?
- What are the needs of the intended users? What are their ‘jobs’, ‘gains’ and ‘pains’?
- Is the issue to be addressed and the potential solution in line with any overarching strategies or objectives?

17 The value proposition canvas can help to identify the needs: [https://www.strategyzer.com/canvas/value-proposition-canva](https://www.strategyzer.com/canvas/value-proposition-canva)
The Mobility Dashboard is a tool developed to make mobility data accessible. The planned solution first collects data in to a "data lake" to allow analysis and modelling on such information. The subsequent step is building an actual dashboard that shows the mobility data and that enables simple analysis, mainly for users to make such data as accessible as possible.

The solution responds to the needs of the citizen by offering a dashboard from which they can access data easily and rapidly, allowing them to make simple analysis. The dashboard furthermore supports decision makers and planners in their work by providing a tool to be able to collect and access data from various sources and make analysis and models in order to provide better mobility services.

The city of Bergen is facing a deep transformation with regards to mobility, moving from a transportation mainly located in the suburbs that exploits private fossil-fuelled cars towards a sustainable and fossil-free flow, based in the core of the city. Thus, the challenge is represented by both rethinking physically the city and adopting new technology to face an uncertain future. Implementing more sustainable modes of transportation is a prioritized issue for making Bergen become fossil free in 2030.

For more information see:
SCORE website
Demo
2. Define solution

When you have identified and agreed on the challenge you want to address, you can start looking at the potential solutions. Most of the challenges that cities face today are not unique for a specific city, the majority are commonly shared. For instance, issues related to traffic flow & routing, mobility hubs, drainage & flooding, and air quality are areas where many cities are experiencing challenges. Innovation is moving fast, and it is possible that if you have identified a common problem in your city, there is another city which has the same problem and which have also been working to develop a solution to manage this. By using personal networks, information sharing platforms, and regional knowledge sharing platforms, you can find tested solutions or components available to be replicated.

When a potential solution is identified there are a number of important questions to ask to assess the potential to replicate and prepare a replication strategy. As an annex to this document you will find a template with questions to ask the solution owner that will help you to assess the potential of replicating a solution in your city. Based on the collected/reviewed information of the original solution you can make an assessment of the level of replicability and define which components have relevance for reuse and the level of adaptation of the solution, or the standard operating procedures in the city, that is needed.

Questions to consider:

- Is there someone who has a similar problem? How have they solved it?
- What is the level of replicability and applicability of the solution that they have?
- Which technical components would be relevant for you to reuse?
- How much adaptation do they need, is this possible to do?
- Are the use cases applicable to your situation?
- Is the solution available with an open source license?
- Is the documentation available, understandable and completely public?
- Are the contextual factors similar to the situation in your city?
- Where do the existing solution overlap and differ from the requirements you foresee at this point in your envisioned solution?

Where to find solutions and components developed by others?

In Europe there are several existing or emerging initiatives that share already developed solutions and components for others to use. Some examples include:

- SCORE solutions
  [www.northsearegion.eu/score/score-solutions/](http://www.northsearegion.eu/score/score-solutions/)

- The Connected Smart Cities & Communities Catalogue, an initiative of Open & Agile Smart Cities network (OASC)
  [www.oascities.org/catalogue/](http://www.oascities.org/catalogue/)

- Eurocities Knowledge Society Forum

- Joinup, funded by the ISA2 Programme of the European Commission

- Open Source Observatory (OSOR)
3. Ensure context

In parallel to defining the solution and the preparation of the technical aspects of the solution it is also necessary to start reviewing and preparing the contextual factors as soon as possible to ensure that the development, testing and implementation will be successful.

Here we identify four main components that help to assure the context: the stakeholders; the legislative and security framework; the compatibility of the digital structure and the business case.

**Figure 6: Four main contextual components**

Get the right stakeholders onboard

Typically, there are four categories of stakeholders to engage: relevant city departments (for instance: urban planning, environment, waste, mobility, information and communication technology, data and information), national and/or local agencies in the same fields as above, relevant private sector actors, and affected end users. First of all it is important to identify who the stakeholders are, how they are affected and what type of confluence they have. Secondly, for real engagement, mechanisms must be in place to assure that all stakeholders are heard and understood. A two-way communication throughout the processes is crucial.

Make sure security and privacy issues are addressed and in compliance with applicable legislation

Offering public services through digital solutions has potential security or privacy issues. Legislation and regulations can sometimes be seen as a limiting factor limiting adoption of new technologies. On the other hand, the legislation around software and use of public data is usually there to ensure security and guarantee privacy of information. Security and privacy aspects will need to be addressed based on the means for data collection and the ways in which data will be used.
Review the business case

The main tool to get the decision makers onboard is a really good business case. The business case for the solution will show the investments needed and the expected returns. Some smart city projects do not have direct return on investment in monetary terms which makes it important to also try to quantify and estimate a social or environmental return in terms of reduced impact on the environment or improved services. Using open source code has the potential to reduce the development costs while offering high quality services.

Make an initial assessment of the compatibility of the digital structure

As mentioned in previous sections the unlocking of city data is central in many smart city solutions. Before initiating a replication process, it is essential to verify that you have access to the necessary data and that it comes in a format compatible with the software. To get the necessary data, it is necessary to assure that the required infrastructure, such as sensors, can be put in place. The new solution must also fit within the digital architecture and procedures of the city.

Questions to consider:

- How do we get to user acceptance and adoption?
- What are the potential legal, privacy and ethical issues?
- How do we assure the short term and long-term financing?
- How will the future solution match the digital architecture, system maintenance and security policies of the city?

Case study: Citizen Science as a Service (CSaaS) - for Flooding

Citizen Science as a Service consists of a flood reporting mobile app that allows citizens to contribute to flood monitoring by reporting flood related events and issues.

The application has been developed by the University of Bradford to allow citizens to collect and provide flooding data. Through the app, the citizens can report issues related to flooding for a specific geographic location and with descriptive texts and photos. The reported information is processed and analysed through intelligent techniques to classify the information and share with local authorities for decision making support.

Since the app is based on the engagement from citizens their involvement is crucial. The app is first developed and tested for Bradford and Aberdeen where the local councils have initiated and organised meetings with local community groups to analyse the communities’ needs and collect feedback on the design of the prototype. Due to the involvement of citizens, the preparation of the context for implementation has also included a process for ethical approval.

For more information see:

https://northsearegion.eu/score/score-solutions/citizen-science-as-a-service-csaas/
For any replication scenario, there will be some adaptations necessary to make the solution ready to be integrated into your city’s digital structure. Typical technical adaptations that are made to customize software to be used in a new context include:

- make parts of code generic
- translations to local language and adapt user interface
- make adjustments to match city software policies
- make adjustments to match other hardware
- data - prepare to collect and/or connect to city data

In terms of city operations and contextual factors, the overall objective is to prepare and nurture the context so that it enables the implementation of a new solution. Looking at how the solution developer has addressed the contextual factors in their city, gives inspiration for how to (re)create the context and assure that the implementation will be possible. If for instance a legal issue was identified, this will need to be addressed in the strategy for replication and possibly in the adaptation of the solution. If adoption by users proved to be a challenge in previous living lab testing, this will have to be addressed both in the stakeholder engagement and potentially also in the adaptation of certain technical aspects. Additionally, a replication process can lead to questioning the way you are currently organised to do a certain task. The new perspectives brought in through the replication may help you to reconsider some practices and trigger innovation and change in your processes.

4. Make adaptations
Potential to test in an urban living lab?

Testing visioned solutions in a living lab or a test bed at an early stage of development can help to understand the usefulness of the solution and the need for adaptation. By using a developed or reused prototype of the solution, first general assumptions on how the solution solves a challenge can be validated (or invalidated). In case of success, the use case tested can become a Proof of Concept (POC). And in that POC you can choose to deploy a living lab approach to validate more specific assumptions with stakeholders as part of the development of the solution.

The benefit of testing in an ULL is that it gives the opportunity to try implementation of a solution in real-life ecosystems, in this case in a city. This means that the solutions will immediately be experienced by end-users in an everyday setting (or realistically simulating one in the lab), be connected to the city systems and undergo feedback rounds with developers.

As specified in the SCORE ULL testing guide, the most important compounds of a ULL are a) integration of residents and other stakeholders (researchers, public organizations and private companies), 2) co-development of innovative ideas, systems and solutions; 3) complex and real-life communities and settings, 4) defined geographical area, and 5) evaluation of actions. When preparing for testing and evaluation of a solution, there are three layers of aspects that should be taken into account:

- Social aspects related with end-users: resources, attitude, knowledge, and personal traits
- Technical aspects related to the developed solution: user friendliness, technical problems, barriers and challenges, usefulness to users (Does it really solve the problem or help the user solve it?)
- Socio-technical aspects related to process: timing and design of the task, interaction between actors and/or developers

More details on living lab testing and the development and preparation of a test case are found in the SCORE ULL Testing Guide.¹⁸


Questions to consider:

- What is the level of customisation needed for the technical components? Are these adaptations possible to make?
- In what way does the context in our city require us to adapt the solution? To which extent are the use cases similar or different in between the original solution and the envisioned solution in our city?
- Is it possible and relevant to adapt any operations in the city to correspond to the original use cases? Or vice versa?
Before making a large-scale implementation it is recommended to test the solution in an (Urban) living lab, a test-bed environment. The identification of a suitable lab for testing and the preparation of a testing plan is to be done in parallel to ensuring the context and adaptation of the solution as discussed in previous sections. The testing of the solution allows us to collect evaluation data and analyze the user experience evaluation. A living lab field test takes around six months from starting the design of the test to making the final evaluation.

The outcome of the testing phase is a thorough evaluation of the solution to improve the solution for further implementation and upscaling. The evaluation of a living lab test includes both technical and conceptual aspects of a solution. To make sure that all relevant feedback is collected and leads to the improvement of the solution, the evaluation should be done through an inclusive process and be well documented. The documentation of testing of a solution contains important information for anyone who is trying to replicate. It may be useful for a replicator to request and review this documentation.

5. Test and prepare for implementation of the replicated solution

Living lab testing step by step

1. Design the task for users
2. Define the timeframe for testing
3. Select the user group
4. Conduct a workshop and baseline evaluation
5. Narrow down the user group
6. Test the solution
7. Ensure continuous data collection
8. Perform mid-term evaluation and implement improvements
9. Close the test period officially and perform final user experience evaluation

Source: SCORE ULL Testing Guide

Example: Masterportal

In 2012 HAM-LGV started to develop the Masterportal which is now used by more than 25 different cities and authorities in Germany and Austria. The idea of the Masterportal originated from a need to reduce the number of map portals with different code bases used in the city of Hamburg. The solution was to create one portal and the apps (with configuration files) using it. The portal would have one shared code to take care of and be available to be replicated to other use cases in the city. With the Masterportal for Hamburg available the replication to other cities in German speaking started. Today the solution is managed through an implementation partnership and a concept for co-development with the implementation partners and support from Dataport (City of Hamburg’s IT provider).

The objective is that, through SCORE, the replication of the solution can be extended to other cities outside of Germany and Austria, for instance, the cities of Ghent and Bradford. To prepare the solution for replication, the code needs to be adapted, so that languages can be added easily via variables and language files. Each function e.g. for the command "print" will get a variable and with the language files these variables can be output in the selected language easily. Additionally, the documentation will be changed to English.

In summary, six different steps have been identified and initiated to internationalise the solution:

1. Add variables to code (done).
2. Create first translation language files
3. English (done), Spanish, Italian, French, Turkish
4. Translate Documentation (https://bitbucket.org/geowerkstatt-hamburg/masterportal/src/dev/doc/)
5. Translate Website (www.masterportal.org)
6. Create a concept for maintenance of the translation

For more information see:
https://www.masterportal.org/
https://northsearegion.eu/score/score-solutions/open-source-geoportal/
Replication of open source smart city solutions has great potential to offer high quality public service delivery through more efficient, collaborative and inclusive software development. However, the idea of taking a complete solution developed in one city and replicating it directly in another city is typically not possible. Solutions built for one specific context must instead be appreciated as smaller components of functionalities before being adapted for use in a different context to create a new solution. If components are clearly defined and made available with complete documentation and source code available under an open source license, other cities, or departments within the same city can reuse what is already available for new use cases to meet their challenges.

From a solution developer’s perspective, there are benefits to gain by sharing and offering the solution you have created as open source freely available to others. With more people depending and working on the same solution, the liability to assure the code works and the interest to correct and further develop it is shared leading to higher quality and innovation. From a replicator’s perspective, there are obvious gains of attempting to replicate what is already available. By replicating, you get access to tested solutions or components without the need to develop new source code. Even though the context in every city is unique, there are generic components developed that can be integrated into the digital architecture to efficiently create new solutions.

→ Replication should be understood as a wider concept of reusing different aspects and components of a full solution. Developing generic technical components that can be easily adapted and applied to multiple use cases make replication efficient.

→ Identifying similarities in data between cities and striving to adopt mechanisms to ensure interoperability of systems can be is a good starting point for replication.

→ A solution developer can facilitate replication by assuring findability, documentation and demonstrations, quality code and accessible repository, and an appropriate open source license.

→ For a replicator, the five main steps to take for replication include: identify need, define solution, ensure context, make adaptations, and test and prepare for implementation.

→ The principles of open source and the aim to facilitate replication promote collaboration and co-creation which can help improve public service delivery, reduce costs and assure transparency and sovereignty.

5. Concluding remarks
These guidelines are part of Work Package 5 in SCORE “Demonstrate and Replicate solutions in nested living labs” led by Johanneberg Science Park and particularly the task 4 “Developing a replication plan and guidance document”.

The document was developed and written by Bax & Company during 2020 with great support from SCORE partners. Special thanks to Tim Van Achte, Digipolis Ghent for good contributions and Justine Ottevaere (Ghent city), Jef Willems (Digipolis Ghent), Hein Wills (City of Amsterdam), Nicole Schubbe & Thorben Finger (City of Hamburg), Bupesh Mishra & Dhaval Thakker (Bradford University), Martin Brynskov & Kata Boronte (Aarhus University), Kim Lantto & Ågot Watne (City of Gothenburg), Eva Pavic (Johanneberg Science Park /IRIS) and Claus Mullie (Foundation for Public Code) for valuable input.