FINANCIAL SERVICES REGULATORY AUTHORITY
سلطة تنظيم الخدمات المالية

Guidance – Application Programming Interfaces (APIs) in ADGM

DATE: 14 October 2019

Table of Contents

INTRODUCTION 3
BACKGROUND 4
Objectives of the API Guidance 5
What is an API 5
The types of APIs 6
REGULATORY REQUIREMENTS 7
Anti-Money Laundering 7
Data Protection 8
Third Party Outsourcing 8
API REQUIREMENTS 9
Design 9
API Documentation 9
Security 10
Cyber security 11
Encryption 11
Two-factor authentication 12
Penetration testing 12
Credentials management 12
Monitor API activity 13
Error handling 13
Data 13
API Governance 13
Version control 13
Depreciation policies 14
APPENDIX 15
Appendix A: API Design Comparison 15
Appendix B: API Standards 16
Appendix C: Technology Standards 17
Appendix D: Data Standards 18

INTRODUCTION

1) This Guidance is issued under section 15(2) of the Financial Services and Markets Regulations 2015 ("FSMR"). It should be read in conjunction with FSMR, the relevant Rulebooks of the Financial Services Regulatory Authority ("the Regulator"), and the Guidance & Policies Manual of the Regulator.

2) This Guidance is applicable to those considering developing or using "Application Programming Interfaces (**APIs**)", including applicants for a Financial Services Permission in ADGM, financial services firms located outside ADGM, and participants in FinTech, RegTech, SupTech[1], amongst others.

3) ADGM encourages Financial Service firms to adopt and promote the use of standardised, "interoperable"[2] and trusted Application Programming Interfaces (APIs) in order to create the means to adapt and update in the context of an increasingly complex and changing business environment, and the rapidly evolving needs of customers.

4) The FSRA encourages a standardised approach to creating, maintaining and governing APIs that will allow the development of innovative financial products and approaches in ADGM that will benefit customers and financial institutions throughout the UAE, the region and further afield. It is the intention of the FSRA to promote experimentation, accelerate implementation of cutting-edge technologies, and speed up industry adoption of customer-focused disruptive ideas, in order to help drive financial inclusion and realise the API economy.

5) Organisations that create APIs will be able to pivot, adopt new ideas and discard old ones quickly. They will be able to iterate their products to keep up with changes in customer behaviour in a timely manner. Investing in the agile development mind-set, and therefore APIs, can give an organisation a competitive edge. Organisations who commit to building a marketplace to trade and settle discrete, understandable, and valuable APIs will be able to accelerate their realisation of the API economy's dividends.

6) To that end this Guidance takes a high level overview of the fundamental elements, standards and considerations that the FSRA deems necessary in providing safe and robust APIs. This Guidance should not restrict the use of APIs; rather, it is there to promote standardised approaches to building and providing APIs, which will be promoted in the ADGM Digital Sandbox.

7) This Guidance is not an exhaustive source of the Regulator's policy on the exercise of its statutory powers and discretions. The FSRA is not bound by the requirements set out in this Guidance and may impose additional requirements to address any specific risks

[1] These terms are used in various ways in the financial services industry. "FinTech" at its broadest incorporates all financial technology. "RegTech" includes those technologies that facilitate compliance with regulations. "SupTech" includes those technologies that facilitate supervision of financial markets and actors.

[2] The API is able to exchange and use information with other APIs, different systems, devices, applications or products to connect and communicate in a coordinated way.

posed by APIs/ API developers. The Regulator is not bound by the requirements set out in this Guidance and may modify this Guidance at its discretion where appropriate.

8) Unless otherwise defined or the context otherwise requires, the terms contained in this Guidance have the same meanings as defined in FSMR and the Glossary (GLO).

9) For more information please contact the FSRA at FinTech@adgm.com

BACKGROUND

10) Advances in new technologies, and maturity of others, have provided opportunities for significant change and disruption to financial services and other related activities globally. Powering this innovation are APIs. APIs can fuel internal innovation, reach new customers, extend products and create vibrant partner ecosystems. APIs by their very nature allow for rapid prototyping, agile development and a fail fast, learn quick culture. They provide a way to share, move and access information previously ring fenced within isolated systems.

11) "Big Tech" companies[3] are opening up access to vast resources and computing power providing access to cutting edge technology, such as machine learning neural networks, blockchain development tools and even quantum computing, that were previously unavailable to the wider market. Additionally, in recent years there has been wide adoption of open-sourced technologies, giving developers suites of tools to create new programmes, systems and networks.

12) Combined with the ever growing surge of the use of smart phones, consumers are now expecting seamless digital interactions tailored to their own specific needs. Which in turn is giving rise to the 'Challenger' or 'Neo' banks who are focused on providing customers with personalised 'experiences' rather than standard financial products.

13) These new business models represent a fundamental step in the evolution of the financial services industry and have already disrupted more traditional ways of offering financial services. For example, 'marketplace banking' business models, i.e. exposing internal digital business assets or services in the form of APIs to external counterparties, is creating an entirely new ecosystem of banking services predicated on intelligent data management and agility in developing new products. The creation of and broadening of access to new data assets are in turn creating many new opportunities for both incumbent and start-up organisations. Fundamental to the development of this new paradigm is the "API economy[4]" which facilitates efficient and secure access to data and processes held at different actors within the financial services sector.

[3] These include some of the world's largest multinational technology corporations, e.g. Google, Apple, Facebook and Amazon.

[4] https://www2.deloitte.com/content/dam/Deloitte/us/Documents/financial-services/us-fsi-api-economy.pdf

14) However in order to realise an efficient API economy, APIs must be able to 'talk' the same language. In recognition of this, several open banking initiatives such as in the UK[5], the EU[6], Singapore[7], Hong Kong[8], Australia[9] and New Zealand have taken this one step further to maximise interoperability and collaboration, by mandating certain Financial Institutions (FIs) to make data available (in the banking sector, often termed "Open Banking" or "Open Data" in a broader context) according to strict standards, predicated upon API usage.

15) While the FSRA does not propose that Open Data or APIs are made mandatory it does see them as an integral part of any FIs digital strategy and as such will look to align with international best practices so as to maintain a safe and trusted digital environment.

Objectives of the API Guidance

The high level objectives of the API guidance are to promote:

a. *Interoperability* - to promote the adoption of globally recognised and accepted standards, to ensure the sustainable growth of the digital economy, interoperability across sectors and connectivity to global markets

b. *Security & Trust* – to promote the use of internationally recognised security and governance practices in order to safeguard consumers and the financial services market.

c. *Innovation* - to drive and encourage a culture of innovation and competitiveness.

d. *Collaboration* - to advance and foster collaboration amongst the financial services and technology ecosystems.

What is an API

16) An API can be seen as a user interface just with different users in mind. Rather than an individual interacting with an application on their smart phone, it is a computer application interacting with another, over the internet or within a private network, using predefined rules described in the API.

[5] https://www.openbanking.org.uk
[6] https://openbankproject.com
[7] http://www.mas.gov.sg/Singapore-Financial-Centre/Smart-Financial-Centre/Financial-Industry-API-Register.aspx
[8] https://www.hkma.gov.hk/media/eng/doc/key-information/press-release/2018/20180718e5a2.pdf
[9] https://treasury.gov.au/consultation/c2018-t247313/

17) Some APIs are designed to enable the query or update of a database, other APIs simply enable a process that has been exposed by one system to be initiated by another. In each API interaction there are the 'providers' of the API and the 'consumers' of the API:

- 'API Provider' refers to an organisation that exposes their data or services through APIs;
- 'API Consumer' refers to any organisation or person who uses an exposed API to access and consume the data or information.

18) In order for a successful interaction between the API Provider and API Consumer, the terms of their engagement (protocols) have to be pre-defined. Once both parties have agreed this so-called 'API Contract', thereby establishing the relevant permissions to connect, then interactions and interoperability can be instantaneous and potentially limitless.

The types of APIs

19) APIs can be classified into to the following three types (although many methodologies for classification exist):

- **Private** APIs – used within an organisation to provide interoperability between internal applications in order to help automation and provide flexibility.
- **Partner** APIs - used to integrate software between a company and its partner, often for a very specific purpose like providing a product or service.
- **Open** APIs - an interface that has been designed to be easily accessible by the wider population where a business relationship is not necessary.

20) In terms of design and governing rules there are currently two widely-used types of API methodologies in the financial services industry (although as of the date of this Guidance, newer approaches such as GraphQL are emerging and should be considered if they are relevant):

- Representational State Transfer (REST); and
- Simple Object Access Protocol (SOAP).

21) To connect different systems and networks, both approaches can leverage the Hypertext Transfer Protocol (HTTP[10]), which defines how messages are formatted and transmitted over the internet, and encryption techniques so that the information being passed cannot be read by anyone other than the originator and the intended recipient. However, the two types differ in terms of structure and approach and as such have different applications in mind.

[10] https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

22) REST is a framework which provides a specific methodology for how to design, build and operate an API which allows an application to use certain commonly-used and standard HTTP operations[11]. These operations enable one application to retrieve, send, update and remove data from another application[12]. RESTful APIs can output data in various different formats. These attributes make RESTful APIs easy to adopt, and flexible in connecting systems of different types.

23) SOAP is another methodology and differs from REST especially in that it only uses one format, XML. SOAP also allows an application to programme another application directly using a wide degree of functions. Given these attributes, and the wide use of the XML standard in financial services, SOAP is like REST a commonly-used API methodology in the industry.

24) SOAP is often used for transactional operations such as in payment gateways. It was developed in order to enable the API Provider to expose business logic to approved API Consumers so that they could interact safely over any communication protocol being used, usually the internet, in order to initiate a specific automated process.

25) REST is often used in situations where rapid, wide-scale adoption is a goal, for example mobile apps for social networks or web chat services. It was developed in order to facilitate simpler information sharing in a fast and efficient manner over HTTP only.

26) The most appropriate type of API style to use will depend on the environment, the project scope, the processes required and the type of information being shared.

27) A more detailed comparison between the two API design styles can be found in **Appendix A**.

REGULATORY REQUIREMENTS

28) Due to the very nature of collaboration and innovation that an API economy encourages amongst the financial services sector and others, the FSRA's expectations regarding API Consumers and Providers and maintaining a safe, sound and trusted financial services ecosystem are set out in this Guidance.

Anti-Money Laundering

29) Money Laundering (ML) and Terrorist Financing (TF) are two major risks that threaten economic growth and social stability through the illicit flow of funds and illegal activities. ML and TF pose significant negative impacts on the financial system.

30) As such the FSRA expects organisations providing or consuming APIs to adhere to the FSRA's Anti Money Laundering and Countering Financing of Terrorism "AML/CFT"

[11] APIs that use the REST methodology are called "RESTful" APIs

[12] These functions corresponds to the GET, POST, PUT and DELETE commands respectively.

framework at all times and put the appropriate measures in place to mitigate these risks, as well as:

a) UAE AML/CFT Federal Laws, including the UAE Cabinet Resolution No. (38) of 2014 Concerning the Executive Regulation of the Federal Law No. 4 of 2002 concerning Anti-Money Laundering and Combating Terrorism Financing;

b) the FSRA AML and Sanctions Rules and Guidance ("AML Rules") or such other AML rules as may be applicable in ADGM from time to time;

c) the adoption of international best practices (including FATF Recommendations); and

d) monitoring national and international sanctions lists.

Data Protection

31) Protecting confidentiality and security of customer data is vital for the stability and reputation of any financial services institution and should not be compromised. As such, organisations are required to comply with the ADGM Data Protection Regime[13] and to apply best-practice safeguards for the security and protection of sensitive customer data during transit, processing and storage.

32) There are also a number of secure hosting standards, e.g. ISO27001, which organisations should adhere to. This standard aids organisations in securing their information and helps implementation of an information security framework that is appropriate to the scale and maturity of the relevant organisation, the services they provide, and the third party suppliers they contract with.

33) For a list of technical standards that should be considered when providing and consuming APIs, please see **Appendices B** and **C**.

Third Party Outsourcing

34) For organisations regulated by the FSRA any issues that may result from the outsourcing including the failure of any third party to meet its obligations are the responsibility of the regulated organisation (GEN 3.3.31, PRU 6.8).

35) In its assessment of a potential third party service provider, the regulated firm must therefore satisfy itself that the service provider maintains robust processes and procedures regarding the relevant service (including, for example, in relation to the testing and security required in this section on Technology Governance).

[13] https://www.adgm.com/operating-in-adgm/office-of-data-protection/overview

API REQUIREMENTS

36) This section is intended to provide guidance on industry best practices around the design, security, maintenance and use of APIs in order to ensure interoperability, resilience and scalability of the API economy that we wish to encourage in ADGM and with other international API implementations.

37) It is recommended that an organisation should first identify why it wants to develop and provide (or consume) an API, who the stakeholders within the organisation are, who the audience for the API will be, the systems and business processes involved and the actors/system that the API will interact with or replace.

Design

38) All APIs should:

a. Have platform independence – any web or mobile client should be able to call and interact with the API, regardless of how the API has been implemented internally.

b. Allow for unhindered API evolution – APIs should flexibly evolve and be able to add functionality independently from other applications using them. As the APIs evolve, the existing applications using them should be unaffected and can continue to function without having to update to the latest version of the API.

c. Use appropriate data standards – APIs should be using the most relevant data standard that are applicable for the type of data being transacted and the use case it is being applied to. Where there is no fit within an existing data standard organisations may decide their own data specification. However, it should publish the associated definitions using a 'data dictionary' in line with industry practices such as those outlined in the Open API Specification or Web Service Definition Language.

d. Have good data security - It is important to have stringent information security, cyber security and other data related policies/guidelines.

e. Be complete and concise – an API should be easy to understand and work with, as should be the API contract. Implementing and integrating with it should be a straight forward process.

API Documentation

39) The API documentation (or 'contract') describes all aspects of the API in order to enable successful interaction between the API provider and API consumer. As such it should be a concise reference manual containing all the information required to work with the API, with details about the functions, classes, return types, and arguments. The API contract should, where relevant, be supported by tutorials and examples.

40) At a high level the fundamentals that need to be documented in the API contract in order for both parties to be able to interact are:

a. The business rules and service agreed between the API Providers and Consumers.

b. The rules around how each party authenticates themselves before gaining access to the API.

c. The standards that the API is adhering to including the change management and version control information that the consumer must be aware of.

d. The design of the API i.e. its structure, the resources (data) that it provides access to and how to interact with the API to obtain them.

e. The certification, on-boarding and acceptance of the API consumer from the API.

41) As such the API contract should also include the following content (but not be limited to):

- sampling code and example responses
- rules on information handling, incident management and risk management
- method of authentication (and how it impacts service interoperability, single sign-on, and rate-limiting)
- design changes (recent and planned) and versioning information
- availability, latency, ownership, depreciation policies and status capability
- approach to backwards compatibility
- guidance on configuring the API to make sure any relevant governance frameworks are followed
- the open data standards used
- security information
- cost of use of the APIs, if applicable
- support that will be provided to the consumer of the API

Security

42) Most important of all the considerations for organisations providing and consuming APIs is the security measures that are deployed, which must comply with network security best practices. Updates and patches to all systems, particularly security systems, should be performed as soon as safely feasible after such updates and patches have been released. The following sections set out the main risk areas and mitigations for these that, in the opinion of the FSRA, need to be taken into account.

43) As a general rule organisations providing and using APIs should also ensure that all parties that they are engaging with:

- Use access tokens to establish trusted identities and control access to the services and resources.
- Encryption and signatures are employed as standard.
- Quotas and throttling are in place that determine how often APIs can be called. For example, more calls on an API may indicate that there is a Denial-of-Service attack. Or it could also be a programming mistake such as calling the API in an endless loop.
- API traffic is enforced using an API gateway that allows authentication as well as control.

44) For more detailed technology standards that should be employed please see Appendix B.

Cyber security

45) As APIs are another entry or exit point for an attack on an organisation, the API security strategy must include the following cybersecurity mitigations (but not be limited to):

- Strong firewall defences
- Vulnerability and threat management
- Antivirus and malware protection
- Denial of Service (DoS) or Distributed Denial of Service (DDoS) protection
- Patch management
- Email filtering
- Web filtering
- Administration privileges
- Access control
- Intelligence and information sharing

Encryption

46) The encryption of data, both at rest and in transit, should be included in the security policy. In particular, encryption and decryption of private keys should utilise encryption protocols, or use alternative algorithms that have broad acceptance with cyber security professionals. Critical cryptographic functions such as encryption, decryption, generation of private keys, and the use of digital signatures should only be performed within cryptographic modules complying with the highest, and ideally internationally recognised, applicable security standards.

Two-factor authentication

47) As well as ensuring that architecture supporting the API and the API itself is secure, organisations should also consider the use of two-factor authentication (2FA) when APIs are initiated by a consumer accessing online service. 2FA is an extra layer of security designed to ensure that the only person who can access an account is the individual who owns it, even if the individual's password has been compromised. The process involves the user providing two different authentication factors to verify themselves.

48) However, it is worth noting that whilst this reduces the chance of being hacked, 2FA is not completely secure and still relies on the vigilance of the individual. For example, phishing attacks purportedly coming from trusted services login page can result in users giving away their credentials. In some extreme cases, hackers have been able to negate 2FA by spoofing an individual's SIM card and intercepting the unencrypted message as it is sent over the network.

Penetration testing

49) It is recommended that all systems and infrastructure should be regularly tested for vulnerabilities by an external penetration testing expert who is professionally accredited (such as CREST, IISP, TIGER scheme or OSCP Offensive Security).

Credentials management

50) Authentication, authorisation and encryption are fundamental to the security of APIs. In terms of authentication, as far as possible, API providers should have a well-defined process to help ensure that individuals or organisations are robustly authenticated.

51) Authorisation should only allow the authorised/accredited organisations and people to have access to the right API resources.

52) Organisations must therefore ensure that they have the appropriate infrastructure in place for secure storage and management of relevant access credentials. These credentials include (but not limited to):

- Identity keys
- Signing keys
- OAuth client IDs and secrets
- Usernames and passwords
- Access tokens

53) Where authentication processes are handed off or redirected to other sites or apps, the technical processes should avoid the potential for disclosure or interception of the credentials. The organisation should also maintain the ability for the user to verify the authenticity of the site into which they are entering their credentials such as displaying the relevant URL and lock icon that they are interacting with.

Monitor API activity

54) The security of an API is only as good as the organisation's day-to-day security processes. All APIs should be monitored for unusual behaviour such as changes in IP addresses or users using APIs at unusual times of the day.

55) It is recommended that the ability to write audit logs before and after security related events is in place as this increases the potential to monitor and detect attacks.

56) Larger organisations should also look to create a Security Operations Centre (SOC) dedicated to monitoring, assessing and defending enterprise information systems such as APIs, web sites, applications, data servers, networks, hardware and software.

Error handling

57) All responses to errors should use the commonly used HTTP codes and should not reveal details of the failure unnecessarily as this may provide unintended attack vectors for bad actors.

Data

To enable the interoperability of APIs at all levels (whether among systems, sectors, or geographies), the adoption of common data standards is necessary. Open data standards and ontologies provide a reference point that enables two parties to share data and information in a way that ensures understanding is preserved and the meaning can be conveyed.

58) To that end organisations should adopt international open data standards and ontologies when providing an API in order to ensure maximum interoperability.

59) For more detailed information on appropriate data standards please see **Appendix D**.

API Governance

60) Business failures have often arisen as a result of the lack of adequate technology-related procedures, including, for example, lack of security measures, systems development methodologies, limited system penetration testing for operating a robust business, and lack of technical leadership and management. The FSRA has therefore included specific Guidance regarding expected controls and processes to help mitigate these issues.

Version control

61) Versioning and change control is very important and needs to be managed effectively. As such, organisations should have formalised policies and procedures in respect of the following where relevant:
    - release numbers for all major and minor releases
    - backwards compatibility for all API changes
    - support for technology developers for major API versions for a specified period

- escalation path for when vulnerabilities come to light
- make a new endpoint available for significant changes

62) If, however, for some reason the change is not backwards compatible, then the organisation must consider:
    - Incrementing a version number in the URL (start with /v1/ and increment with whole numbers).

    - Supporting both old and new endpoints in parallel for a suitable time period before discontinuing the old one.

Depreciation policies

63) Clear API depreciation policies should be in place so old client applications are not unnecessarily supported.

64) The time by which users/consumers have to upgrade, and how they will be notified of these deadlines should be clearly stated.

APPENDIX

Appendix A: API Design Comparison

65) The following table describes the main differences between the SOAP and REST API design styles.

SOAP	REST
SOAP is function driven and focuses on exporting pieces of application 'logic' rather than data. It relies exclusively on XML to provide messaging services.	REST is data driven and is focused on accessing named 'resources' (which represent data or an object) through a single consistent interface.
SOAP uses a Web Service Definition Language (WSDL) that describes the functionality offered by a web service for communication between the consumer and provider.	Most Web services using REST can obtain the needed information using a URL. REST uses four standard HTTP operations (GET, POST, PUT, and DELETE) to perform tasks on the resources.
SOAP only permits output in XML.	REST can output data in many different data formats e.g. XML, Comma Separated Values (CSV and JavaScript Object Notation (JSON).
SOAP has successful/retry logic built in.	There is no inbuilt 're-try'; the client has to re-try if the process fails, as REST is stateless.
SOAP is good for: • Enterprise services • High reliability and security • Asynchronous processing • Large data sets • Interacting with legacy systems	REST is good for: • Web services • Limited bandwidth (smaller messaging sizes) • Limited resources (no XML parsing required) • Exposing data over the internet • Combining content from many different sources
Common use cases: • Financial services • Payment gateways • Telecommunication services	Common use cases: • Social media services • Social networks • Web chat services • Mobile service

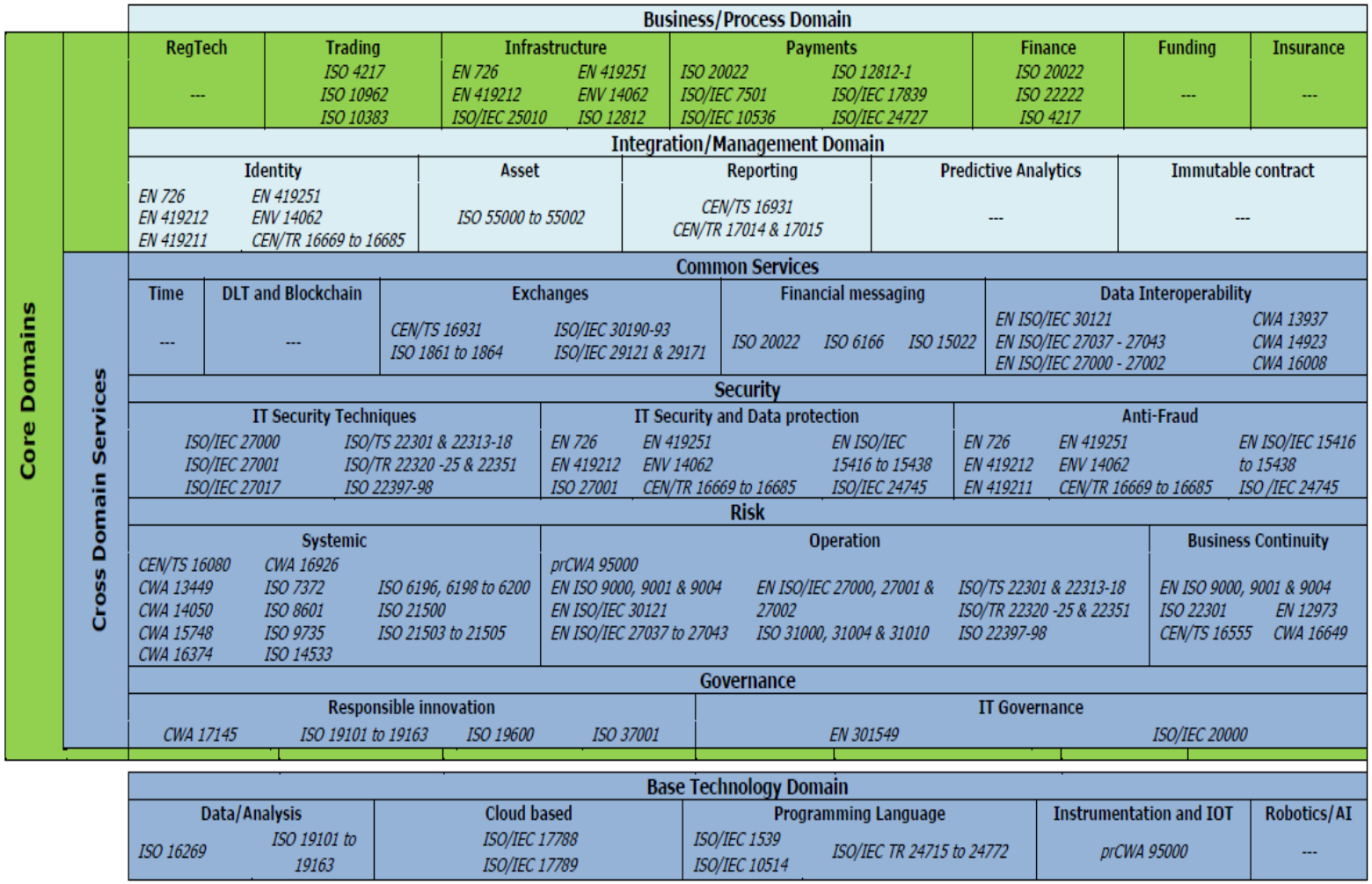
Appendix B: API Standards

66) The following describes the standards that should be applied when building APIs.

Area/Subject	Standard
Publishing of technical, engagement details and data dictionary	At a high level the API provider should provide the API documentation and a data dictionary. Code samples, tutorials and a software development kit (SDKs) should also be provided. Documentation and definitions should be in line with industry practices outlined in the Open API Specification (Swagger), the RESTful API Modelling Language specifications (RAML), or the Web Service Definition Language (WSDL) for SOAP APIs.
API architecture/Communication Protocols (i.e. the code to call the API)	RESTful (Representational State Transfer) or SOAP (Simple Object Access Protocol) depending on the use case. Firms should provide for conversion of SOAP to REST or vice versa if relevant.
Message/Data format (i.e. how the API response/payload will be provided)	JSON (JavaScript Object Notation) or XML (eXtensible Markup Language) depending on the use case. With field names based on ISO 20022 where relevant.
Transmission of data	HTTPS & TLS v1.2 (Transport Layer Security)
Onboarding of customer onto service	username/password and two-factor authentication where appropriate
Authorisation	OAuth 2.0 N.B RESTful API calls with the HTTPS protocol should use a session-based authentication using OAuth 2.0 and JSON web tokens (JWT)
Authentication	SAML 2.0 or OpenID
Encryption	AES, RS, SHA 256-bit where relevant depending on the use case

Appendix C: Technology Standards

67) The following is a list of European and International standards, where available, that should be considered when building APIs.



Appendix D: Data Standards

68) The following describes some of the international data standards that should be applied where relevant for APIs in order to ensure interoperability.

Open data standard/Ontology	Area/Use
Financial Industry Business Ontology (FIBO) https://www.omg.org/hot-topics/finance.htm	Defines financial industry terms, definitions and synonyms using semantic web principles such as OWL/RDF and widely adopted OMG modeling standards such as UML. Providing a means for integrating disparate technical systems and message formats, and aid in regulatory reporting by providing clear and unambiguous meaning of data from authoritative sources.
Financial Industry Regulatory Ontology (FIRO) https://github.com/GRCTC/FIRO	FIRO is a series of interlinked Ontologies based on industry standards to capture regulatory imperatives and rules in formal semantics. It will enable efficient access to, and smarter consumption of, the wide and complex spectrum of financial services industry regulations.
Financial Regulatory Data Format (FIRE) https://github.com/SuadeLabs/fire	The Financial Regulatory data format defines a common specification for the transmission of granular data between regulatory systems (in finance)
International Financial Reporting Standards (IFRS) https://www.ifrs.org/	Provide a common global language for business affairs so that company accounts are understandable and comparable across international boundaries.
ISO20022 https://www.iso20022.org/	Standard for electronic data interchange between financial institutions. It describes a metadata repository containing descriptions of messages and business processes, and a maintenance process for the repository content. The standard covers financial information transferred between financial institutions that includes payment transactions, securities trading and settlement information, credit and debit card transactions and other financial information.
eXtensible Business Reporting Language (xBRL) https://www.xbrl.org	Business documents, such as financial statements, performance reports, or compliance reports. The standard formats allow the documents to be transmitted and parsed between entities easily

Open data standard/Ontology	Area/Use
Statistical Data and Metadata eXchange (SDMX) https://sdmx.org	Designed to describe statistical data and metadata, normalise their exchange, and improve their efficient sharing across statistical and similar organisations.
Open Financial Exchange (OFX) http://www.ofx.net	Open Financial Exchange is a reference data standard used for exchanging financial data, and performing transactions between financial institutions and underlying applications.
Association for Cooperative Operations Research and Development (ACORD) https://www.acord.org/standards-architecture/acord-data-standards	ACORD is the global standards-setting body for the insurance and related financial services industries.
Financial products Markup Language (FpML) http://www.fpml.org	FpML is a standard based on XML and used for data exchange for electronic dealing and processing of derivatives instruments like interest rate derivatives, inflation swaps, dividend derivatives and other structured products.
Financial Information eXchange (FIX) http://www.fixtradingcommunity.org	FIX is the standard used for pre-trade and trade messaging across Financial Markets globally. It describes trade-related messages, and used for automated trading of securities, derivative, and other financial instruments.
Market Data Definition Language (MDDL) http://xml.coverpages.org/mddl.html	MDDL enables the exchange of information necessary to account, analyse, and trade financial instruments.