Supporting Telecom Business Processes by means of Workflow Management and Federated Databases[♣]

Wim Nijenhuis and Willem Jonker
Service Development and Support
KPN Research
PO Box 15000, 9700 CD Groningen
The Netherlands
r.w.nijenhuis@research.kpn.com
w.jonker@research.kpn.com

Paul Grefen
Computer Science Department
University of Twente
PO Box 217, 7500 AE Enschede
The Netherlands
grefen@cs.utwente.nl

Abstract

This report addresses the issues related to the use of workflow management systems and federated databases to support business processes that operate on large and heterogeneous collections of autonomous information systems. We discuss how they can enhance the overall IT-architecture. Starting from the OSCA architecture, we develop an architecture that includes workflow management systems and federated databases. In this architecture, the notion of information systems as a monolithic entity disappears. Instead, business processes are supported directly by workflows that combine presentation blocks, function blocks, and data blocks. We address the specific issues of transaction management and change management in such an architecture.

[♣] A short version of this report is to appear as a paper in the proceedings of the Association for Information Systems 1997 Americas Conference, Indianapolis, Indiana, August 15-17, 1997.

Table of Contents

1. Introduction

Core telecom business processes such as network planning, network management, and service management operate on large and heterogeneous collections of autonomous information systems. Since requirements on these systems have changed over time, the systems have been adapted quite often and consequently serve quite different demands than those they were originally designed for. The systems have become large monoliths that are difficult to maintain and have lost the flexibility to support (ever changing) business processes effectively.

Both the distributed systems community (e.g. ODP, TINA) and the information systems community (e.g. federated databases, workflow management) work on the development of architectures, techniques, and tools to address these problems. For instance TINA (Telecommunication Information Networking Architecture) defines an open software architecture for telecommunication services. The TINA DPE (Distributed Processing Environment) seperates the functional applications and physical infrastructure (e.g. switches) and thus enables the use of heterogeneous systems. Also, federated databases enable the uniform access of heterogeneous databases.

For application developers it is often difficult to understand how these architectures, techniques, and tools can be brought together to solve their problems. We have come accross this problem several times inside the business units of our company. The main problem is how to use new technologies such as workflow management and federated databases in an overall information system architecture.

Within Dutch PTT, the overall system architecture is based on an architecture similar to the OSCA architecture [Bell92]. Therefore, we present our approach on integrating workflow management and federated databases into the OSCA architecture. In the extended architecture, monolithic information systems as such disappear and business processes are supported by workflows. This report does not discuss the integration in depth but instead shows the basic changes to the architecture and discusses the benefits.

1.1 Structure of this report

Before discussing the architecture we propose, we start with an exploration of related work in Section 2. Next, we introduce the standard OSCA architecture in Section 3. In the next two sections, we present our extensions to this architecture: in Section 4, we add federated database systems and in Section 5, we add workflow management systems. In Section 6, we discuss the advanced aspects transaction management and change management in the context of the proposed architecture. We end this report with some conclusions.

2. Related work

This report proposes a modular, layered architecture to cope with the changing infrastructure and functionality of complex information systems, taking into account recent developments in the fields of federated databases and workflow management.

In the past, several approaches to flexible information systems architectures have been described. An important class is formed by approaches based on the concept of building blocks that provide ‘chunks’ of encapsulated functionality with clear interfaces. This class is closely connected to the ideas from the object-oriented software engineering world. A recent example of this class is found in an approach towards structuring telecommication systems [Lind95]. A second important class is formed by approaches based on layering the functionality of systems. The best-known examples are probably the the ISO-OSI 7-layer reference model for network systems [Stall87] and the ANSI-SPARC three-schema architecture for database systems [Tsich78]. In information system architectures, middleware layers have been proposed to act between information servers and user clients, e.g. the mediator approach [Wied92]. In the present report, we combine the building block and layering approaches using federated database systems and workflow management systems as basis for the infrastructural layers.

In the field of federated database systems or multidatabases, a number of system aspects relevant to this report have been investigated recently. Architectures for multidatabase systems have been described allowing for the integration of multiple databases, see e.g. [Bukh96]. Traditional basic transaction mechanisms, as originally described in e.g. [Gray81], have been extended to deal with the characteristics of federated systems. An example is the work in [Gref96], where transactional protocols are proposed for integrity control in federated database environments.

In the field of workflow management systems, attention has been paid to reference architectures. A well-known architecture is that of the Workflow Management Coalition [WMC94]. The work in [Gref95] proposes a more detailed reference architecture that allows for modular extension. Advanced transaction models like nested transactions [Daya90, Daya91], cooperative transactions [Elma92], and long-running transactions [Garc87] are important ingredients to obtain transactional workflows [Agra93].

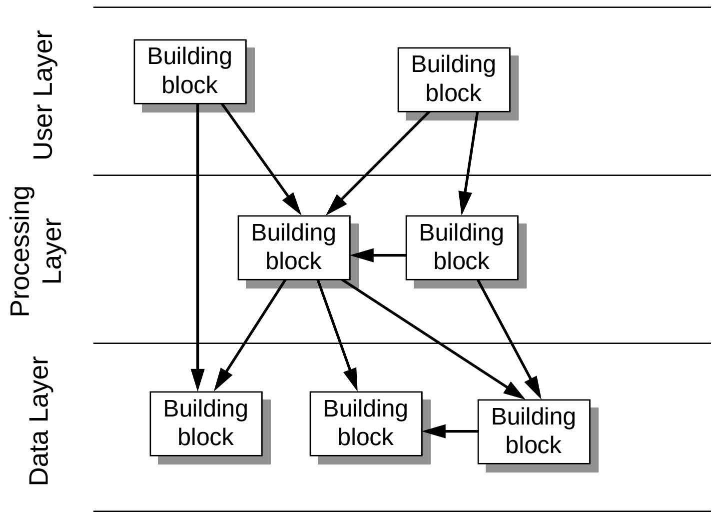

Figure 1: The OSCA architecture.

3. An architecture for information systems

The OSCA architectur is well known within the Telecoms industry. We briefly introduce the OSCA architecture [Bell92]. The key point of OSCA is to distinguish three different layers in the architecture of information systems: the data layer, the processing layer and the user layer. Each layer consists of building blocks.

A building block is a set of computer programs, data schemas, and other related software. The functions it provides to other building blocks are described in interfaces. Other building blocks need no knowledge about the internal structure. This way it offers the flexibility to combine software products without defining an architecture for the programs. The involved parties have to agree on the individual interfaces though. This requires an unambiguous definition for instance by using CORBA-IDL [OM95].

With the separation in layers, the architecture is based on the concept of separation of concerns. The aim of OSCA is to enable reuse of common data and common functionality. For example by introducing a data layer and separating the data from the application, it becomes easier to share data among different applications.

- The **data layer** groups the corporate data management functionality. This layer has the most stable structure over time. The corporate data is strongly related to the high level business strategy. What are our products? Who are our customers or partners? etc. This information is managed by the data layer and has a company wide scope. Changes in the architecture of the data layer are typically related to business changes (new product, services, types of customers). It is important to observe that there is private data (opposed to corporate data) in all three layers. Private data is owned by a building block. An example of a building block in this layer could be “Client”: a building block that is responsible for the data related to the company’s clients.
- The **processing layer** contains business operations and management functionality. The processing layer is based on the enterprise model. How are things done? What actions are required before there is a contract with a customer? Business functions are executed by the processing layer and often require access to corporate data in the data layer. Changes in the

processing layer are typically related to a fundamentaly different way of doing the same business, i.e. the same level of change as Business Process Engineering (BPR) efforts often try to achieve. An example of this could be the move from selling on a pre-paid basis to charging afterwards. Also a shift to a more customer-oriented focus often results in changes in the processing layer. An example of a building block in this layer could be “Provide Facility”: a building block that is responsible for giving clients access to communication facilities like call forwarding.

- The **user layer** contains the human interaction functionality. The user layer is based on the human business tasks and goals. It serves as an agent for carrying out the user’s tasks by providing functionality and accessing functionality of other building blocks. So the user layer is strongly determined by the way users want to or are supposed to perform their tasks. For example, sending letters, entering a new customer, selling a product. The user layer is the only layer that has interaction with the user, i.e., the user can only access business functionality or corporate data via the user layer. Changes in the way users perform their tasks results in changes in the user layer. This may happen almost every day. Therefore flexibility is of great importance to the user layer. An example of a building block in this layer could be “Serve Client by Phone”: a building block that supports employees who answer incoming calls.

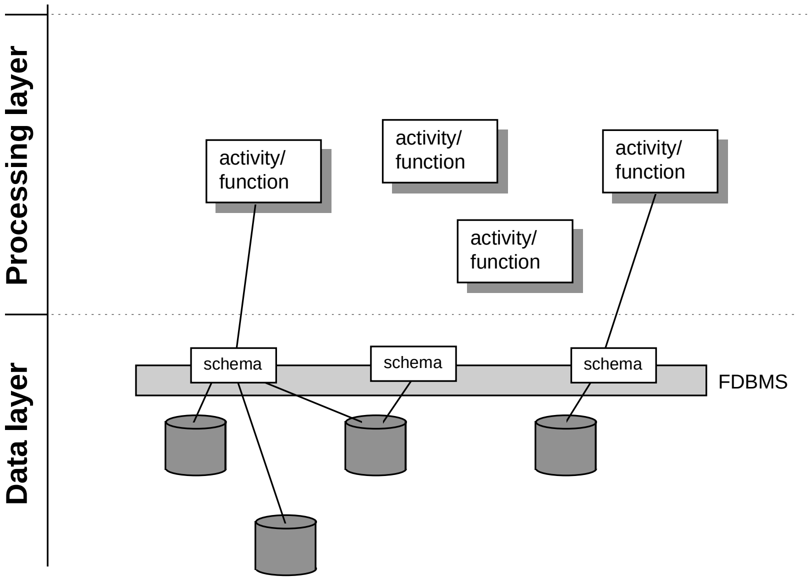

Figure 2: FDBMS in OSCA

4. Positioning federated databases

By separating data from an application, OSCA makes corporate data available to other applications. The next question to be answered then is how to access these corporate data, i.e. how do user and processing layer building blocks access data layer building blocks? To obtain a maintainable architecture it is preferred to have uniform access to building blocks. A popular way to obtain uniform access to databases is to use gateway technology (e.g. SQL gateways, ODBC). However, this technology only deals with syntactic uniformity and fails to address the problem of semantic heterogeneity of the data layer building blocks. Semantic heterogeneity denotes differences in representation of the same or similar information in different databases.

The field of federated databases addresses the problem of uniform access to collections of heterogeneous databases. A federated database management system[1] (FDBMS) is a 'virtual' DBMS on top of the collection of heterogeneous databases, and it provides specific mechanisms to cope with both syntactic and semantic heterogeneity. Federated databases provide support for:

- schema integration as a means to deal with syntactic and semantic heterogeneity
- distributed transaction management over multiple autonomous DBMSs
- query optimization over multiple DBMSs

Research into federated databases was originally motivated by the problem of legacy systems (see [Brod92, Shet90]). However, we argue that database federation is a good structuring principle of its own, and as a result it is a useful technology in structuring newly developed information systems.

[1] In this report, we use the term 'federated database system' to indicate a tightly-coupled heterogeneous collection of local database systems. Often the term 'multidatabase system' is used for this kind of system to distnguish it from 'federated database system' indicating a loosely-coupled collection of local database systems.

In the OSCA architecture, we position a FDBMS in the data layer as a uniform access mechanism for accessing the data layer building blocks. This means that we force the higher level building blocks to access corporate databases through the FDBMS. This simplifies the implementation of building blocks in the processing layer and user layer. The functionality added by a FDBMS makes it easier to:

- offer a database schema that is tuned to the needs of a user layer or processing layer building block;
- coordinate transactions over multiple data layer building blocks;
- improve performance of access to multiple data layer building blocks.

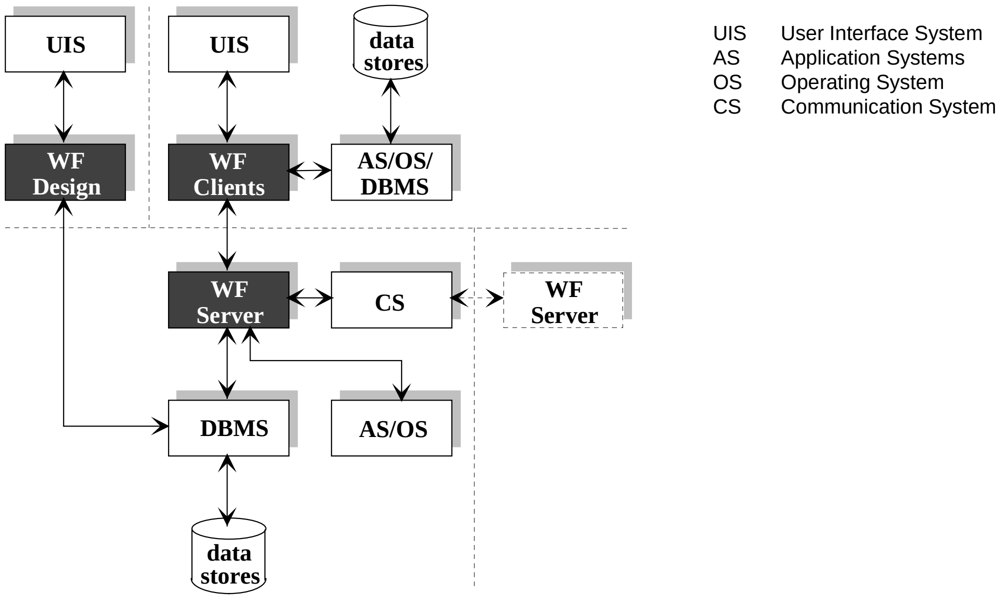

Figure 3: WFMS reference architecture from [Gref95]

5. Positioning workflow management systems

In the previous section, we have showed how the basic OSCA architecture can be extended with federated databases to provide uniform access to the data building blocks. In this section, we show how we can further extend the architecture with workflow management tools to *provide uniform access to the processing layer building blocks*.

WFM systems typically provide support for the following aspects relevant in the context of this proposal:

- uniform access to business applications on heterogeneous systems;
- control over processes involving multiple autonomous information systems;
- automatic performing of process steps by executing appropriate applications.

We position workflow management systems based on the reference architecture in [Gref95]. In this architecture, workflow client, workflow server, and database interface functionality are clearly distinguished. Coupling with (legacy) applications is positioned at the server level (usually batch applications) or at the client level (usually interactive applications). Coupling between multiple workflow management systems is positioned at the server level. Coupling with the underlying multidatabase can be structured according to the various data sets distinguished in [Gref95]: organization data, product data, schema data, process data, management data, and application data.

When relating workflow systems to the OSCA-architecture, they can be seen as the means to specify and control the use of business applications (i.e. processing layer building blocks) and corporate data (i.e. data layer building blocks). Thus WFM offers a way to specify the rules related accessing different business applications and corporate data. In current IT systems, business rules are either implemented separately for every application or left to the responsibility of the user. Positioning WFM in the OSCA architecture enables locating business rules within the architecture.

We position WFM in the processing layer as a uniform access mechanism for accessing the processing layer building blocks, i.e. we force the user layer building blocks to access applications through the WFM engine (Figure 4). Thus a WFM engine enhances the architecture by enabling an implementation for the other building blocks that is more or less independent of a specific business process. The WFM engine coordinates the use of business functions implemented by processing layer building blocks even if they are on heterogeneous systems. This simplifies the implementation of building blocks in the user layer.

In current IT systems, business rules are either implemented separately for every application or left to the responsibility of the user. For example, the sales support function can ask for a solvency check if the contract exceeds certain predefined thresholds. The business rule related to the solvency check may be hard coded in the concerning application or left to the users knowledge of business rules specified in the process definition. Using a WFM layer in the extended architecture, business rules can be specified and enforced by the workflow management system.

Summarizing, we see the following benefits in the extended architecture:

- separate position for business rules avoids the need for changes in processing layer building block for reasons other than changes in functionality;
- it becomes easier to incorporate legacy systems in the architecture, the WFM engine caters for integration problems of applications (just as a FDBMS does for data);
- (support for) process control is identified as a core activity of IT. This gives way for better IT-support for managing processes.

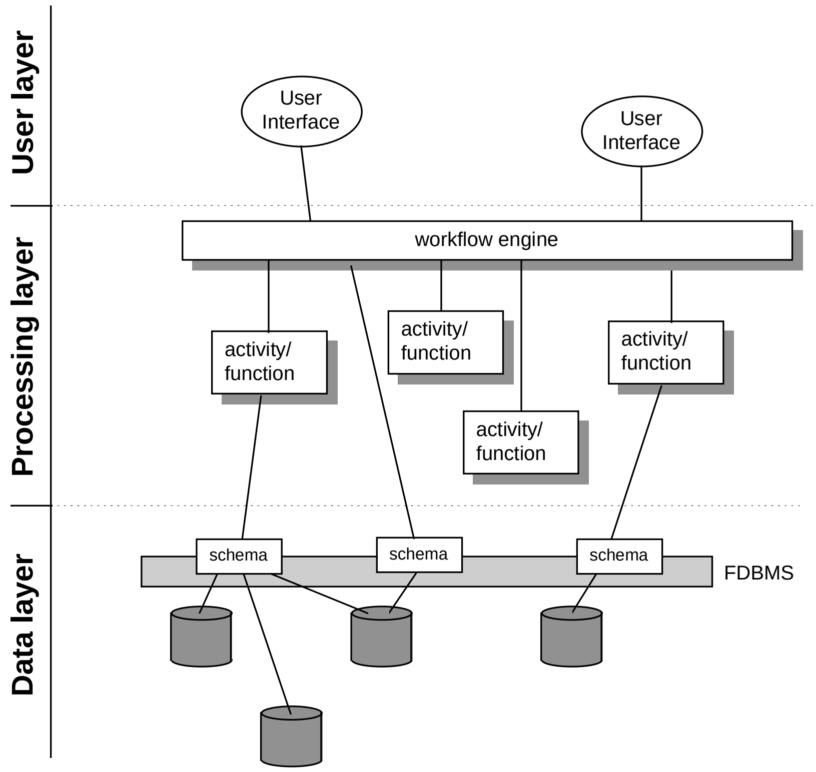

Figure 4: WFMS in OSCA

6. Advanced aspects

In this section, we discuss two advanced aspects in the context of the extended OSCA architecture discussed in the previous sections. First, we describe how high-level business transaction management can be handled in the architecture. Next, we discuss change management, i.e. dealing with system evolution, in the context of the architecture.

6.1 Transaction management

The architecture we propose is a layered architecture consisting of a federated database system that provides a uniform access to the corporate databases and on top a workflow management system that provides uniform access to the common functions. An important issue to be addressed is how to support transactions in such an environment. Both the database community and the workflow community work on new transaction models that can support advanced applications (see [Elma92]). For our layered architecture, we suggest a two-level transaction model: a basic transaction model supported by the federated database management system, and an advanced transaction model supported by the workflow manager. The basic transaction model should support the traditional ACID properties and could be a 2PC/2PL protocol or a nested transaction model. The advanced transaction model should be implemented on top of the basic transaction model and should support typical long-lived business process transactions. A similar approach is described in [Gref97].

Note that this approach also puts a natural limit on the complexity of the function building blocks. Since these building blocks access the data through the federated database system, their transactions are limited to those supported by the federated database system. More complex business process transactions should be supported by combining function building blocks by means of the workflow manager.

6.2 Change management

We briefly illustrate the flexibility of the proposed architecture with respect to change by examining the actions to be taken upon the following changes:

- change of a data building block
- change of a function building block
- change of a business process

We consider some possible changes of a data building block. First the data building block can be ported to a new implementation platform (for example a hierarchical database is replaced by a relational database) without changing the information content. Such a change will be completely invisible to the function building blocks, since this is completely dealt with in the federated database. Also in the case of merging data building blocks or splitting them, there will be no effect on the function building blocks as long as the information contents remains the same. Note that function building blocks would have to be modified in case of direct access to data building blocks. Changes that involve change of information content (e.g. schema evolutions) may affect function building blocks. To what extent depends on how much of the change can be covered in the federated database and the information need of the function block.

Changing a function block will only affect those workflows that contain the specific function block. To what extent the workflow description needs to be changed depends on the level of change. Another option that eliminates the need of changing the workflows is by introducing a new modified function block while leaving the old one in place.

Change of a business process will lead to a change of the workflow description for that process. Depending on the availability of the required data and functions, new data building blocks and function building blocks may be needed.

As one can see, the architure always limits change to the building blocks involved. Change is always in terms of building blocks and not change in terms of entire applications or entire information systems.

7. Conclusions

We have presented an extension of the OSCA architecture that includes workflow management tools and federated databases. By doing so, we have shown how these new technologies can be incorporated to enhanche an existing information infrastructure. The resulting architecture extends the OSCA architecture by adding uniform access to collections of data builiding blocks as well as to collections of function building blocks.

In addition, we have outlined how mixtures of basic database transactions and advanced business process transactions might be supported in a 2-level transaction model. Finally, we have illustrated the flexibility of the architecture with respect to changes in data, functions and processes.

We are well aware that quite a few issues have to be solved in order to realise the archictecture described above, we list some below:

- the development of practical advanced transaction models;
- the interaction between transaction models in a layered architecture;
- the interaction in case of multiple workflow managers;
- the development of design tools.

Nevertheless, we experience that the current commercially available tools allow medium scale experimentation with the principles outlined in this report.

8. References

[Agra93] R. Agrawal, D. Lomet (Eds.); *Special Issue on Workflow and Extended Transaction Systems*; IEEE Data Engineering Bulletin, June 1993.

[Bell92] Bellcore, *The Bellcore OSCA Architecture*, Technical Reference TR-STS-000915, Issue 1, October 1992.

[Brod92] Brodie, M. L., *The Promise of Distributed Computing and the Challenges of Legacy Systems.* In: Gray, P. M. D.; Lucas, R. J. (eds.): . LNCS 618, Springer-Verlag, Berlin, 1992, pp. 1-28.

[Bukh96] O.A. Bukhres, A.K. Elmagarmid; *Object-Oriented Multidatabase Systems*; Prentice Hall, Englewood Cliffs, NJ, USA, 1996.

[Ceri84] S. Ceri, G. Pelagatti; *Distributed Databases: Principles and Systems*; McGraw-Hill, New York, USA, 1984.

[Daya90] U. Dayal, M. Hsu, R. Ladin; *Organizing Long-Running Activities with Triggers and Transactions*; Proceedings 1990 ACM SIGMOD International Conference on Management of Data; Atlantic City, USA, 1990.

[Daya91] U. Dayal, M. Hsu, R. Ladin; *A Transactional Model for Long-Running Activities*; Proceedings 17th International Conference on Very Large Databases; Barcelona, Spain, 1991.

[Elma92] A.K. Elmagarmid (ed.); *Database Transaction Models for Advanced Applications*; Morgan Kaufmann; San Mateo, California, USA, 1992.

[Garc87] H. Garcia-Molina, K. Salem; *Sagas*; Proceedings 1987 ACM SIGMOD International Conference on Management of Data; USA, 1987.

[Gray81] J. Gray; *The Transaction Concept: Virtues and Limitations*; Proceedings 7th International Conference on Very Large Data Bases; Cannes, France, 1981.

[Gref95] P.W.P.J. Grefen, R.N. Remmerts de Vries; *A Reference Architecture for Workflow Management Systems*; Memorandum INF95-36; University of Twente, 1995; submitted for publication.

[Gref96] P. Grefen, J. Widom; *Protocols for Integrity Constraint Checking in Federated Databases*; Proceedings of the 1st International Conference on Cooperative Information Systems; Brussels, Belgium, 1996.

[Gref97] P. Grefen, J. Vonk, E. Boertjes, P. Apers; *Two-Layer Transaction Management for Workflow Management Applications*; CTIT Technical Report 97-07; University of Twente, 1997.

[Lind95] F.J. van der Linden, J.K. Müller; *Creating Architectures with Building Blocks*; IEEE Software, November 1995.

[OM95] Object Management Group; *The Common Object Request Broker: Architecture and Specification, Version 2.0*; Object Management Group, 1995.

[Shet90] Sheth, A.; Larson, J.: Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys, Vol. 22, No. 3, 1990, pp. 183-236.

[Stall87] W. Stallings; *Handbook of Computer Communications Standards, Vol. 1*; Macmillan; New York, USA, 1987.

[Tsich78] D. Tsichritzis, A. Klug; *The ANSI/X3/SPARC DBMS Framework*; AFIPS Press, 1978.

[Wied92] G. Wiederhold. Mediators in the architecture of future information systems. IEEE Computer, 25(3):38-49, March 1992.

[WMC94] Glossary - A workflow Management Coalition Specification; Workflow Management Coalition; November 1994.