

## Research Article

# Flexible Investment Strategies for Cloud-Native Architecture of Public Health Information Systems

**Ming Jiang**<sup>1</sup>, **Ichiro Nakamoto**<sup>1</sup>, **Weiqing Zhuang**<sup>1</sup>, **Weiguo Zhang**<sup>1</sup>, **Yin Guo**<sup>1</sup>,  
and **Liting Ma**<sup>2</sup>

<sup>1</sup>School of Internet Economics and Business, Fujian University of Technology, Fuzhou 350000, China

<sup>2</sup>Fujian Jiangxia University, Fuzhou 350108, China

Correspondence should be addressed to Ming Jiang; 19842158@fjut.edu.cn

Received 10 October 2020; Revised 12 November 2020; Accepted 16 December 2020; Published 19 January 2021

Academic Editor: Hongju Cheng

Copyright © 2021 Ming Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The randomness of public health events requires that the cloud-native architecture, as the mainstream architecture of the new generation of the public health information system, has the appropriate flexibility to meet the needs of environmental change. The flexible acquisition of cloud-native architecture requires organizations to invest additional resources. How to plan and formulate resource input is a topic of common concern for public health management and information systems. According to the commercial characteristics of the public health system based on cloud-native architecture, this paper systematically analyzes the external major impact factors and auxiliary factors that affect the flexible cost investment strategy of cloud providers and combines flexible investment strategies to build a cloud-native cost investment model. Finally, case data in practice is applied into the model, and cost planning is discussed according to different situations. The findings indicate that (1) the more cloud providers adopt the changed flexible strategy, the more conducive it is to reduce costs; (2) the larger the application load, the more cloud providers need to use flexible strategies to lower costs; (3) the less the impact of changing the flexible strategy on costs, the more conducive cloud providers use the flexible strategy to decrease costs; (4) the more uneven the distribution of diversity, the higher proportion of investment increases than the proportion of investment, and the more cloud providers consider the investment using flexible strategy. The results of the discussion provide a reference for public health organizations to use flexible strategies and change flexible strategies in a timely manner and expand the research scope of information system cost investment.

## 1. Introduction

In the prevention and treatment of COVID-19 in 2020, the public health information system plays an important role [1]. The temporary influx of many businesses has brought huge challenges to the architectural flexibility of public health information systems with cloud-native architecture as the mainstream. How to invest in architectural flexibility is an urgent problem in the field of information systems and information management research [2]. For this reason, many scholars have conducted a lot of research on the flexible cost investment of information system architecture, but their research results are mainly concentrated on the public health

system of the traditional SOA architecture [3]. There is little research on the public health system with cloud-native architecture as the mainstream. It does not consider the penalty cost of breach of contract with different percentages of normal service time. This article fully considers the nonsystem and unstructured flexible cost input process of public health information system cloud providers with cloud-native architecture as the mainstream and sets the penalty costs caused by default of different normal service time percentages in the decision-making model [4]. In this case, the cloud provider architecture flexible cost investment provides a reference strategy. The main contributions of this article are mainly focused on the first time that the default factors of

different normal service time percentages are considered in the flexible cost of the cloud provider's architecture, and the actual public health system cloud provider case data is used to verify these two aspects. It is supplemented by research related to the field of information management [5].

## 2. Flexible Cost Input Model of Cloud-Native Architecture

The flexible cost of cloud-native architecture is the cost that cloud providers invest in the flexible development of cloud-native architecture, making the architecture flexible to cope with internal and external changes and ensuring the effective operation of the system [6]. The flexible cost investment strategy of cloud-native architecture is a way for cloud providers to control development costs and maximize their benefits and is affected by external factors and auxiliary factors [7].

*2.1. External Impacting Factors.* The commercial service mode of public health information systems based on cloud-native architecture can be regarded as a new type of multitenant information technology outsourcing [8], and cloud providers deliver cloud services to consumers. The investment strategy of cloud providers on the flexibility of cloud-native architecture is affected by the needs of cloud consumers, or the needs of cloud consumers to process their own business are the external influencing factors of cloud providers' flexible cost strategy. Cloud consumers' own business needs mainly include three aspects: business process [9], application load [3], and normal service percentage, of which business process also includes two aspects: process uncertainty and diversity. The external influencing factors affecting the flexible strategy of cloud providers are described in detail as follows [10].

*2.1.1. Process Uncertainty.* As a new type of information system architecture, cloud-native architecture is mainly used for the processing of the business process of multiple tenants (cloud consumers). There is uncertainty in the business process executed between a single and multiple cloud consumers, which can be described as the difficulty to predict the tasks and resources required for the business process in a specific single instance. The uncertainty of the business process can be either external or structural [11]. External environment uncertainty refers to the uncertainty of exogenous input variables. For example, the business process required for the service confirmed by SLA (Service Level Agreement) is heterogeneous for different cloud consumers, and the business process required for a single cloud consumer to perform a task is not the same too. Structural uncertainty refers to the gap between the predetermined tasks and resources demanded by the cloud provider to perform the business process in a specific instance and the actual needs to perform for the services agreed by SLA, such as the execution order of the prediction preparation before the service container may be different from that required by the cloud consumer, and the more management business processes required by the cloud consumer, the greater the structural uncertainty [12]. Therefore, the external uncertainty caused

by the resources required by the service business process requested by cloud consumers, the uncertainty between the preset resources and the actual needs of internal prediction, and judgment will affect the flexible strategy formulation of cloud providers, and the uncertainty of the process will affect the flexible cost strategy of cloud providers [13].

*2.1.2. Process Diversity.* Different cloud consumers or individual cloud consumers need differential tasks to perform a certain business process. The probability of tasks being performed is different, and there is diversity in the process. When the tasks required by cloud consumption to execute a certain business process are similar, there are only a small number of different types of tasks in the whole business process execution process, which is manifested as low process diversity [14]. However, when the types of tasks requested by cloud consumers to perform similar business processes are very different or the business processes performed by a single cloud consumer require a variety types of task execution and each task is executed at almost the same frequency, it is manifested as high process diversity. Cloud providers need to develop different strategies and resources to ensure the quality of services agreed by SLA according to the diversity of business processes required by cloud consumers at various levels; therefore, process diversity directly affects the formulation of flexible cost strategies for cloud providers [15].

*2.1.3. Applied Load.* Cloud-native architecture is an architecture that supports multiple tenants. The application load of cloud services required to support cloud-native architecture is different at each time point, and the application load brought about by each cloud consumer at different time points is also different [16]. Applied loads can be divided into fixed applied loads, cycle applied loads, lifetime applied loads, and unintended applied loads. The characteristics of fixed application load are that in a certain range, its demand for resources is basically the same over time. This type of application load usually does not need to add or remove resources to cope with changes, only needs to initially configure the necessary resources and provide a certain excess allocation rate, and can be automatically processed through the adaptive characteristics of cloud-native architecture [17]. The characteristics of periodic application load are that some cloud consumers are engaged in the main business with a large number of periodic business processes that need to be completed in the same time interval; the peak and usual application load gap is very large and periodic. The lifetime application load is a special case of periodic application load, which is characterized by that it can only appear once in a period, usually accompanied by the occurrence of an event or task, and will be predicted. However, the application load brings great harm to the cloud-native architecture due to the long-term low and sudden application summit of a cloud provider's resource preparation. Unexpected application load is the generalization of periodic application load, for example, when new products of cloud consumers receive unexpected attention, it often brings about a sudden increase in application load, which cannot predict the peak and duration of resource use in advance. To cope with different

application loads, cloud providers need to be equipped with different resources to cope with the changes brought about, that is, application loads can directly affect the development of flexible cost strategies for cloud providers.

*2.1.4. Percentage of Normal Service Time.* Cloud-native architecture takes cloud service as the delivery mode and faces multiple tenants. Different cloud consumers have different requirements for effective service time of cloud-native architecture due to their own business, and the limitation required for the completion of business process processing, which is usually expressed in the form of a percentage of normal service time. On the one hand, cloud providers need to prepare different resources according to the actual percentage of different normal services of cloud consumers to adapt to the changes; on the other hand, when the percentage of normal service time agreed by each cloud consumer cannot be guaranteed, cloud providers need to pay corresponding fees to cloud consumption for the failed service commitment. The percentage of normal service time directly affects the resources that cloud providers need to prepare, that is, the percentage of normal service time directly affects the formulation of flexible cost strategies for cloud providers.

In summary, process uncertainty, process diversity, application load, and percentage of normal service time as external influencing factors directly affect the input decision of cloud providers on the flexible cost of cloud-native architecture. Only by fully considering these influencing factors, cloud providers are conducive to meeting the needs of cloud consumers, making reasonable flexible cost investment decisions, and effectively planning the flexible cost investment of cloud-native architecture to produce benefits.

*2.2. Auxiliary Factors.* In addition to the influence of external factors on the flexible strategy of cloud providers, the flexible cost input cycle and flexible cost input stage are affected by other factors, which are collectively referred to as auxiliary factors. The investment of the flexible cost of cloud-native architecture involves the whole service life cycle. The length of the predetermined life cycle of cloud-native architecture directly determines the time length of investment of cloud provider in flexible cost of cloud-native architecture and affects the investment strategy of cloud provider in flexible cost of cloud-native architecture. At the same time, cloud providers adopt the way of continuous delivery and continuous deployment for the flexible development and maintenance of cloud-native architecture. There are stages for the investment of the flexible cost of cloud-native architecture, and there are problems of time discount rate for the cost of investment at different times, which directly affects the total investment cost of cloud providers, thus directly affecting the investment strategy of cloud providers for the flexible flexibility of cloud-native architecture.

*2.3. Flexible Strategy.* Architecture is the soul of cloud-native application system. The investment cost strategy of cloud providers on the flexibility of cloud-native architecture, that is, the implementation strategy of cloud provider strategy level, directly determines the cost investment of cloud pro-

viders on the flexibility of cloud-native architecture in the future. Due to its importance, different scholars focus on it, for example, some scholars put forward top-down planning mode emphasizing policy consistency and bottom-up planning mode emphasizing compliance with actual operation from the perspective of enterprise strategy formulation mode; some scholars believe that enterprises should plan flexible cost investment strategy according to the way of coping with internal and external environmental changes, such as static flexible strategy using established scheme to cope with changes and dynamic flexible strategy changing according to actual change needs during operation. To better analyze the flexible cost investment of cloud-native architecture, the following describes the flexible strategy of cloud providers based on the flexible cost strategy proposed by Gebauer and Schober [2] in combination with the commercial characteristics of cloud-native architecture.

*2.3.1. Use of Flexible Strategy.* Based on research by Gebauer and Schober. the use of flexibility is here defined as making it possible for cloud-native applications to change within the expected range according to internal and external changes during operation through the expected design and implementation of the flexibility of the cloud-native architecture without violating the requirements of SLA. The following five factors are mainly considered for the flexible design of cloud-native architecture when cloud providers develop the use of flexible strategy:

- (1) Service capacity refers to the service subdomain that constitutes the cloud provider as agreed by SLA and needs to provide cloud consumer services. Cloud-native architecture is a domain-driven design architecture that focuses on the division of some services, for example, a basic e-commerce system can be divided into commodity subdomain, membership subdomain, payment subdomain, order subdomain, and logistics subdomain. The service needs to be divided according to the predetermined capacity
- (2) The scope of the underlying database refers to the analysis capacity and the number of reports that can be provided by the data warehouse. According to the prediction processing capacity, select the appropriate deployment mode and set the corresponding data processing capacity, realize distributed processing, and meet the requirements of normal service time percentage agreed by SLA
- (3) The user interface refers to the different characteristics and methods that the cloud-native architecture can provide for the user to interact with the system, for example, the HTML5 and CSS are used to realize the support for multiple browsers
- (4) Processing capacity refers to the number of users that the cloud-native architecture can accommodate at the same time as well as the number of transactions and user requests processed

- (5) Adaptive capacity refers to the scenario where the cloud-native architecture can automatically allocate the application load within the scope of the SLA agreement and link each subservice domain. Therefore, in the design, the application presentation layer provides the user-defined interface and defines the input and output modes through visual means as far as possible; the service layer is loosely coupled, forms the software architecture with container technology as the core based on micro-service, provides the workflow definition, realizes the flexible transaction model, and supports the intelligence of the application software; the active database is mainly used in the data processing layer; the system software layer should remove the dependence on the hardware through the virtual and can obtain the hardware resources as needed

*2.3.2. Change in Flexible Strategy.* According to Gebauer and Schober, the difference between using flexibility and changing flexibility is that using flexibility strategies is that the system can be modified, adjusted, and reinstalled through predesigned service capabilities, databases, user interfaces, and processing capabilities, and adaptive flexibility before significant problems occur without causing system usability problems or only causing minor problems to the system. Changing flexibility is conceptually related to changes in system architecture, as long as it is measured by the amount of effort required to change a given system after initial implementation. It generally includes the modularity of the system, such as the splitting of original services, data integration, and related human resource cost investment. Sien and Zhenyu refer to changing flexible strategy as a dynamic flexible strategy, which mainly refers to the dynamic evolution and dynamic refinement of software architecture. In this paper, the change of flexible strategy mainly refers to the continuous delivery and continuous deployment of cloud-native architecture by cloud providers within the requirements of SLA after the cloud-native architecture is online, such as the continuous change of service module and the continuous deployment of service container.

*2.3.3. Manual Flexibility Strategy.* Cloud Provider requires service SLA agreement for the whole service process of cloud consumers. In case of failure to use flexible strategy and change flexible strategy to provide services satisfying SLA requirements within the scope of SLA agreement, cloud provider will adopt an emergency response plan, which requires additional fees. Generally, while manually responding to internal and external changes and restoring services, due to illegal SLA agreements, compensation needs to be paid to cloud consumption. So when cloud providers adopt manual flexible strategies, the flexible costs incurred by them include the additional cost of manual repair of recovery services and the compensation cost to cloud consumers.

### 3. Outline Model of Flexible Cost Investment of Cloud-Native Architecture

In summary, external influencing factors and auxiliary factors will affect the flexible investment strategy of cloud-native architecture of cloud providers, and the investment strategy of cloud providers to the flexible investment strategy of cloud-native architecture determines the investment of cloud providers in the flexible cost of cloud-native architecture, which determines the total cost invested by cloud providers in the whole life cycle of cloud-native architecture. Based on the aforementioned points, a summary model of flexible cost investment in cloud-native architecture is proposed in Figure 1.

### 4. Computational Model for Flexible Cost Input of Cloud-Native Architecture

According to Gebauer and Schober's research method on the flexible cost generation mechanism of information systems, the actual decision-making process of cloud providers is obtained, as shown in Figure 2 below. By integrating the external influencing factors, auxiliary factors, flexible strategy, and flexible cost of cloud-native architecture in the outline model, the computational model is obtained according to the decision process in Figure 2 to better describe and discuss the proposed outline model. Generally, the investment of cloud providers in cloud-native architecture is divided into two stages: design and operation, and cloud providers will have different investment strategies for different stages. In the design stage, first of all, the cloud provider will start this item according to its own strategic decision. If this project is initiated, in addition to the most basic predetermined functions realized by the input cost, the cloud provider decides to adopt the appropriate flexible strategy according to the possible external environment and its own development needs after go-live, mainly including the adaptation of flexible strategy and the change of flexible strategy. In the operation phase, different results will be produced according to the previous flexible strategy and input proportion.

TCOST represents the total cost for the cloud provider to meet SLA requirements throughout the cloud-native architecture lifecycle. It incorporates the cost for a cloud provider to adopt the flexible strategy, the cost for a cloud provider to adopt a changed flexible strategy, the cost for the cloud provider to maintain the system at the operation stage of the system, the cost for the cloud provider to upgrade the system within the operation stage plan, the manual repair cost for the cloud provider to maintain stable manual emergency repair within the time limit during the operation stage, the compensation cost for cloud consumer due to inability to fulfill the commitment within the time limit during the operation stage, and the total cost for the cloud provider to fulfill SLA commitment and make the system return to normal service state.

In summary, the total cost that cloud providers need to pay throughout the lifecycle of the cloud-native architecture



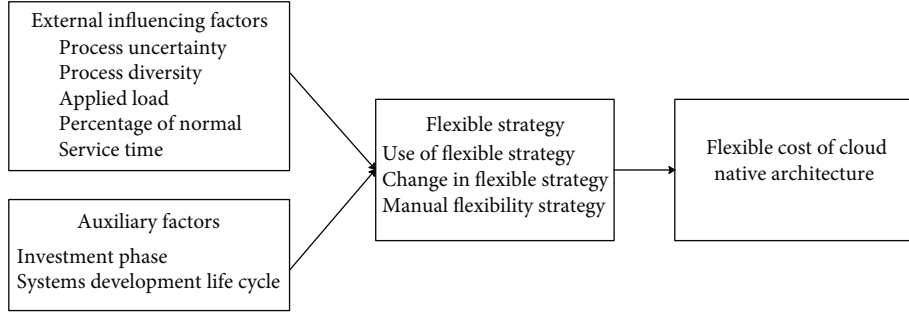


FIGURE 1: Overview model of the flexible cost of cloud-native architecture.

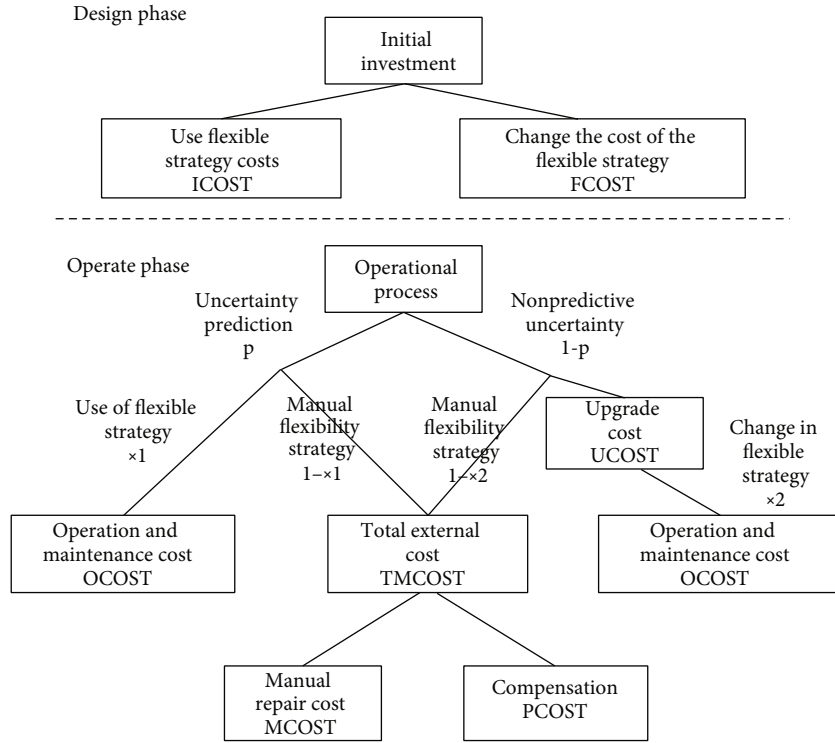


FIGURE 2: Decision process for flexible cost of cloud-native architecture.

is calculated as follows:

$$TCOST = ICOST + FCOST + UCOST + OCOST + MCOST + PCOST. \quad (1)$$

Since

$$TMCOST = MCOST + PCOST, \quad (2)$$

the above calculation formula can be simplified:

$$TCOST = ICOST + FCOST + UCOST + OCOST + TMCOST, \quad (3)$$

and each calculation item in the calculation formula is described in detail below.

(1) The cost when cloud providers adopt the use of flexible strategies during the design stage is governed by the following:

$$ICOST = \{a + bL(x_1)[q + (1 - q)DC]\}z \quad (4)$$

$a$ : the fixed cost required to realize the basic functional requirements of cloud consumers for cloud providers, such as the base class library, authentication, and information management and storage

$b$ : adopt flexible strategies for cloud providers to cope with the costs required for changes, such as using HTML5 to transform the front-end, to realize that there is no need to make a lot of modifications to the architecture when it runs within a predetermined range

$q$  : proportion of cloud-native architecture development that has been completed so far for cloud providers;

$L(x_1) = x_1^\nu [1 - (1 - x_1^\nu)^{1-\nu}]$ : describe the cumulative percentage of tasks that are defined by the cloud provider when the diversity is and the actual execution proportion of the use strategy adopted by the cloud provider is, where the function is proposed by Gebauer and Schober [2], and the curve is time-convex, per the requirements of the Lorenz coefficient curve, and is used to describe the Lorenz curve. It indicates the corresponding relationship between the cumulative percentage of defined tasks and the cumulative percentage of actual tasks performed, and the parameter  $\nu$  ( $0 \leq \nu \leq 1$ ) indicates the degree of curvature of the curve. The greater the concentration of variable distribution, the larger the difference is.

$i$ : represents the annual average interest rate

$T$ : represents the planned life cycle of the cloud-native architecture

$DC = ((1+i)^T - 1)/(i(1+i)^T T)$ : it is proposed by Gebauer and Schober [2] to represent the average discount rate of equivalent cash flow in the lifecycle of a cloud-native architecture

$$\begin{cases} y \in [0, 1] \\ \text{During the design phase, the cloud provider chooses to change the flex policy when } y = 1 \text{ and } y \geq x_2. \end{cases} \quad (7)$$

In the runtime phase, the cost to the cloud provider to upgrade the portion of the cloud provider in the cloud-native architecture design phase with a changed flexible strategy is determined by the following:

$$UCOST = eL(x_2)DC \quad (8)$$

$e$ : the cost of upgrading the part of the cloud provider that uses the changed flexible strategy at the design stage

$L(x_2) = x_2^\nu [1 - (1 - x_2^\nu)^{1-\nu}]$ : represents the cumulative percentage of tasks completed as defined by the cloud provider when the actual proportion of tasks is implemented

$DC = ((1+i)^T - 1)/(i(1+i)^T T)$ : as in Equation (4), it represents the average discount rate of equivalent cash flow in the life cycle of cloud-native architecture

$i$ : resolves the plan life cycle representing the annual average interest rate and cloud-native architecture as in Equation (4)

- (3) The cost of operating maintenance by the cloud provider for the cloud-native architecture design stage using the flexible strategy part and the part upgraded using the changed flexible strategy is given by the following:

$$OCOST = Sd[(q + (1-q)L^{-1}(0.6))w_1 + 0.4w_2]TDC \quad (9)$$

$d$ : it is the cost of the cloud provider to use the flexible

$z$ : is a binary number to ensure that Equation (4) is meaningful and that the following conditions are met:

$$\begin{cases} z \in [0, 1]; \\ x_1 + x_2 > 0, \text{ if } z = 1, \text{ else } z = 0 \\ z \geq 0.5(x_1 + x_2) \end{cases} \quad (5)$$

- (2) The cost when cloud providers adopt the change flexible strategy during the design stage is as follows:

$$FCOST = cy \quad (6)$$

$c$ : it is the cost required for cloud providers to adopt a flexible strategy to change

$y$ : It is a binary number to ensure the validity of the calculation formula (6), which satisfies the following conditions:

strategy part and the part upgraded by the changed flexible strategy for operation and maintenance in the cloud-native architecture design stage

$S$ : it is the amount of stretching of the applied load in the running phase

$T$ : as described in Equation (4), represents the planned life cycle of the cloud-native architecture and the average discount rate of equivalent cash flow in the life cycle of the cloud-native architecture, respectively

$q$ : proportion of cloud-native architecture development that has been completed so far for cloud providers

Since Mutschler (2006) proposed that 60% of tasks with alterations during software running can be done using flexible strategies, 40% of tasks are done by changing flexible strategies [5]. So 0.6 and 0.4 are directly substituted into Equation (9). It represents the percentage of tasks actually performed when the cloud provider defines that the accumulated ratio of completion was 0.6.

$w_1 = px_1$ : it indicates the proportion of cloud providers using flexibility to process the changes required in the operation stage, where it indicates the uncertainty of the task and indicates the proportion of cloud providers using the use strategy to invest in the operation stage

It indicates the proportion of cloud providers that need to change the operation stage by changing the flexibility, in which it indicates the uncertainty of the task and indicates the proportion of cloud providers that use the change strategy to invest in the operation stage.

- (4) In the operation stage, the cost that cloud providers cannot cope with changes requiring manual repair using both flexible strategies and changing flexible strategies is as follows:

$$\text{MCOST} = SfTDC[(1-q)(1-L^{-1}(0.6))w_1 + 0.4w_2 + w_3]. \quad (10)$$

$S, q, w_1, w_2, T$  and  $DC$  are outlined in Equation (9), which, respectively, represent the amount of expansion and contraction of the application load, the proportion of cloud-native architecture development, the proportion of changes required in the operation stage using flexible processing, the proportion of changes required in the operation stage using flexible processing, the planned life cycle of cloud-native architecture, and the average discount rate of equivalent cash flow in the life cycle of cloud-native architecture.

$w_3 = 1 - w_1 - w_2$ : represents the proportion of cloud providers that handle changes using a flexible strategy and a method other than changing flexible strategy handling

$f$ : represents the cost to be paid by the cloud provider for manual repair

- (5) In the operation stage, the cost that cloud providers should compensate cloud consumers for the changes due to their inability to agree on access within SLA is as follows:

$$\text{PCOST} = SrgTDC[(1-q)(1-L^{-1}(0.6))w_1 + 0.4w_2 + w_3]. \quad (11)$$

$S, q, w_1, w_2, w_3, T$  and  $DC$  as in Equation (10), which represent, respectively, the amount of expansion of the application load, the proportion of cloud-native architecture development, the proportion of changes required to the operating stage using flexible processing, the proportion of changes required to the operating stage using flexible processing, the proportion of changes handled using methods other than flexible strategy and flexible strategy processing, the planned life cycle of the cloud-native architecture, and the average discount rate of equivalent cash flow in the life cycle of the cloud-native architecture.

$g$ : indicates that the cloud provider shall compensate the cloud consumer for the cost incurred due to violation of SLA

$r$ : represents the percentage of normal service time, such as the SLA agreed by Amazon and Microsoft, which is 10% of the total price exemption if the percentage of normal service time is lower than 99.9% and higher than 99%, and 30% of the total price exemption when the percentage of normal service time is lower than 99%. The probability of default with 99% normal service time percentage of committed cloud consumers is low, and the probability of default with 99.9% normal service time percentage of committed cloud consumers is high.

- (6) In the operation stage, the total cost required for the cloud provider to fulfill the SLA commitment and make the system return to normal service status is as follows:

$$\text{TM COST} = S(f+rg)TDC[(1-q)(1-L^{-1}(0.6))w_1 + 0.4w_2 + w_3]. \quad (12)$$

$S, q, w_1, w_2, w_3, T$ , and  $DC$  as in Equation (10), represent, respectively, the amount of expansion of the application load, the proportion of cloud-native architecture development, the proportion of changes required to the operating stage using flexible processing, the proportion of changes required to the operating stage using flexible processing, the proportion of changes handled using methods other than flexible strategy and flexible strategy processing, the planned life cycle of the cloud-native architecture, and the average discount rate of equivalent cash flow in the life cycle of the cloud-native architecture.  $g$  and  $r$  in Equation (11) represent the transformed values of the cost paid by compensation cloud consumers and the percentage of normal service time, respectively.

In summary, the total cost will vary according to the cloud provider pair, and different investment proportions, and the size of the decision, and as the primary decision variable for the decision at a fixed time. The above relevant formulas are summarized as follows:

$$\text{TCOST} = \text{ICOST} + \text{FCOST} + \text{UCOST} + \text{OCOST} + \text{TM COST}, \quad (13)$$

$$\begin{cases} \text{TM COST} = \text{MCOST} + \text{PCOST} \\ \text{ICOST} = \{a + bL(x_1)[q + (1-q)DC]\}z \\ \text{FCOST} = cy \\ \text{UCOST} = eL(x_2)DC \\ \text{OCOST} = Sd[(q + (1-q)L^{-1}(0.6))w_1 + 0.4w_2]TDC \\ \text{MCOST} = SfTDC[(1-q)(1-L^{-1}(0.6))w_1 + 0.4w_2 + w_3] \\ \text{PCOST} = SrgTDC[(1-q)(1-L^{-1}(0.6))w_1 + 0.4w_2 + w_3], \end{cases} \quad (14)$$

$$\begin{cases} y \geq x_2 \\ z \geq 0.5(x_1 + x_2) \\ x_1 + x_2 > 0, z = 1 \text{ else } z = 0 \\ y, z \in [0, 1] \\ 0 \leq x_1, x_2 \leq 1. \end{cases} \quad (15)$$

## 5. Analysis of Planning Strategies

To analyze the input planning strategy for the flexible cost of cloud-native architecture of cloud providers in various situations, here, the real cloud-native architecture project case data are applied to the flexible costing model of cloud-native architecture proposed above. Focusing on the goal of minimizing cloud provider cost, the influence of various

external influencing factors on the flexible cost of cloud-native architecture is discussed, to obtain the flexible input optimization strategy set of public health information system based on cloud-native architecture.

*5.1. Data Source.* The data is from the Public Health Information System of a technology company in Xiamen. The system aims to improve health manager demand diversification, drug store management, and information control. Millions of community in China face data analysis, data storage, and many other problems and development. The purpose is to build a collaborative digital supply chain platform to fulfill the connection between traditional community and CDC. The system uses HTML5 and other technologies to implement the front-end display by supporting multiple browsers. For the request of the Web, Nginx is used to process a large number of concurrent requests at the same time by utilizing both multiprocess and asynchronous mechanism. In the background, zuul gateway is used to perform the dynamic loading, compilation, and operation of filters and realize the dynamic connection. The domain-driven design is adopted, and the system is divided into authority certification microservice, basic microservice, configuration center microservice, gateway microservice, account microservice, channel microservice, merchant microservice, payment microservice, functional microservice, questionnaire microservice, and information push microservice, MQ microservice, and timing microservice, to achieve the functional cohesion modularity, the direct connection between service and service through service governance, and the compatibility of interface and deployment environment by using container technology. Finally, the system adopts a Zipkin distributed tracking system to track, identify, and feedback the line state of the operating system in.

*5.2. Data Collection.* The whole project adopts the mode of phased investment. The project start-up time is 2017, the system design and development interval is 1 year in the first stage, and the subsequent delivery and deployment of the system are carried out in the second stage. It has been conducted for 2 years, the project budget cycle is 5 years, and the system architecture is expected to be redesigned and developed within this time period. Therefore, the auxiliary factor system stage, life cycle, and annual discount rate are using bank current interest rate.

Based on the experience of the initial 2 years, the enterprise architect believes that the tasks of using flexible strategies in the operation process of the project are similar to those of using changing flexible strategies and roughly estimates the uncertainty. Because the use of cloud-native architecture in clear accounts is a multitenant system, the tasks of each cloud consumer and a single cloud consumer vary greatly and roughly estimate the diversity; clear accounts need to be used by cloud consumers all over the country, with a relatively high application load and roughly estimated. The conversion value of the percentage of normal service time refers to the reference value proposed by Amazon and Microsoft in their proposed SLA when the percentage of normal

service time is 99.9%; the percentage of normal service time is 99%.

The enterprise architect uses the COCOMO II model proposed by Clark et al. [6] to estimate the number of person-day invested in various work. The specific cost estimate is based on the person-day estimate multiplied by the average salary per person-day. Here is a cost estimation based on the data of the 2019 catch-up network, with an average salary of 300 yuan per person-day.

It is roughly estimated that in the design stage, the basic investment using flexible strategy is adopted for the cloud-native architecture, such as the construction of basic background and the realization of basic functions, with the input cost of 10,000 yuan; in the design stage, the flexible strategy is adopted for the cloud-native architecture, such as DB subdatabase table and DB replication model, with the input cost of 10,000 yuan; in the design stage, the flexible strategy with change is adopted for the cloud-native architecture, such as the construction of continuous integrated release environment, with the input cost of 10,000 yuan; in the operation stage, the flexible strategy with change is adopted to upgrade the cloud-native architecture to cope with the changes, such as issuing a new version of characteristic service, with the input cost of 10,000 yuan; in the operation stage, the flexible strategy with the use and change should be adopted to cope with the operation and maintenance cost of 10,000 yuan for all changes; in the operation stage, the investment cost of manual repair should be 10,000 yuan; in the operation stage, the SLA needs cannot be met, and the total compensation cost is equal to the cost of realizing basic functions.

*5.3. Results and Discussion of the Calculation Model.* To clarify the influence of each external influencing factor and flexible strategy on the flexible cost of cloud-native architecture, MATLAB2017b is used below to calculate the minimum cost of cloud providers in different cases, and the flexible cost investment of cloud-native architecture is discussed by comparison.

*5.3.1. Percentage of Different Normal Service Time.* To clarify the impact of different percentages of normal service time on the flexible cost of cloud-native architecture, here, the minimum value in Equation (3) is solved in four cases according to different flexible strategies of cloud providers. These four cases are the minimum value when the dynamic flexibility strategy is not used, the minimum value when the dynamic flexibility strategy is used, the minimum value when the dynamic flexibility strategy is used, the minimum value when the dynamic flexibility strategy is not used, and the minimum value when the dynamic flexibility strategy is used. The results are shown in Table 1.

According to the above calculation results, the following conclusions can be obtained.

- (1) When the flexible strategy is changed, the minimum value 302.4 is less than the case when not changed, of which the minimum value is 346.9. Comparatively, when the flexible strategy is changed, the minimum value 317.6 is less than the case when not changed,



TABLE 1: Solving units for different percentages of normal service time. Unit: ten thousand yuan.

$a = 30, b = 150, c = 5, d = 10, e = 80, f = 56, g = 30$			
$p = 0.5, v = 0.6, S = 1.4$			
Use flexible strategy $w_1$	Change flex strategy $w_2$	External flex strategy $w_3$	TCOST
$r = 0.1$			
36.5%	0	63.5%	346.9
36.5%	41.6%	21.9%	302.4
$r = 0.3$			
38.6%	0	61.4%	371.8
38.6%	43%	18.4%	317.6

of which the minimum value is 371.8. The results show that investment in changing flexible strategies is beneficial for cloud providers to reduce investment costs regardless of the percentage of normal service time determined by SLA

- (2) In the scenario where the flexible strategy was changed with the minimum of 302.4, the cloud provider input cost was reduced by 44.5. In contrast, when the flexible strategy was changed with the minimum 317.6, the cloud provider input cost was reduced by 54.2. The results show that the percentage of normal service time required by cloud consumers is high, and the more cloud providers adopt the changed flexible strategy, the more conducive it is to reduce costs

**5.3.2. Different Application Loads.** To clarify the effect of different application loads on the flexibility cost of cloud-native architecture, the minimum value in Equation (3) is solved here for four cases: application load at a time with or without dynamic flexibility strategy. The results are shown in Table 2.

According to the above calculation results, the following conclusions can be obtained.

- (1) When no flexible strategy change was used, the minimum value was 267.4, which is in contrast to the minimum value of 346.9 when used. The proportion of investment using flexible strategies increased from 29.4% to 36.5%, and it can be seen that the larger the application load, the more cloud providers need to use flexible strategies to reduce costs
- (2) When the flexible strategy was changed, the minimum value was 245, and this is in comparison with the minimum value of 302.4 when no flexible change occurred. As the application load increased from 1 to 1.4, the proportion of investment using flexible strategies increased from 29.4% to 36.5%, indicating a growth of 7.1%; in the case of changing flexible strategies, it increased from 36.1% to 41.6%, implying a growth of 5.5%. It can be seen that the proportion of investment using flexible strategy increases faster than the case using changing flexible strategy, and cloud providers can consider the investment using flexible strategy more

**5.3.3. Different Uncertainties.** To clarify the effect of different application loads on the flexibility cost of cloud-native architecture, the minimum value in Equation (3) is solved here for the four cases of uncertainty in time with or without dynamic flexibility strategy, respectively. The results are shown in Table 3.

According to the above calculation results, the following conclusions can be obtained.

- (1) When the flexible strategy was not changed, the minimum value was 267.4, which was larger than the minimum value of 240.4 when the flexible strategy was changed. The proportion of investment using flexible strategy increased from 29.4% to 51.5%, from which it can be predicted that the more uncertainty, the more cloud providers need to use flexible strategy to reduce costs
- (2) When the flexible strategy was changed, the minimum value was 235.7; in contrast, when the flexible strategy was not changed, the minimum value was 245. When the uncertainty was increased from 0.5 to 0.7, the proportion of investment using flexible strategies increased from 29.4% to 51.1%, suggesting a growth of 21.7%, and the proportion of investment changing flexible strategies decreased from 35.1% to 34.8%, indicating a reduction of 1.3%. It can be seen that the less the impact of changing the flexible strategy on costs, the more conducive cloud providers use the flexible strategy to reduce costs

**5.3.4. Different Diversity.** To clarify the effect of different application loads on the flexibility cost of cloud-native architecture, the minimum value in Equation (3) is solved here for the four cases of uncertainty in time with or without dynamic flexibility strategy, respectively. The results are shown in Table 4.

According to the above calculation results, the following conclusions can be obtained.

- (1) No flexible strategy change was accompanied by a minimum value of 267.4; comparatively, flexible strategy change was followed by a minimum value of 281.6. When diversity was increased from 0.4 to 0.6, the proportion of investment using flexible

TABLE 2: Solving units for different application loads. Unit: ten thousand yuan.

$a = 30, b = 150, c = 5, d = 10, e = 80, f = 56, g = 30$			
$p = 0.5, v = 0.6, r = 0.1$			
Use flexible strategy $w_1$	Change flex strategy $w_2$	External flex strategy $w_3$	TCOST
$S = 1.4$			
36.5%	0	63.5%	346.9
36.5%	41.6%	21.9%	302.4
$S = 1$			
29.4%	0	70.6%	267.4
29.4%	36.1%	34.5%	245

TABLE 3: Solving unit of different uncertainties. Unit: ten thousand yuan.

$a = 30, b = 150, c = 5, d = 10, e = 80, f = 56, g = 30$			
$S = 1, v = 0.6, r = 0.1$			
Use flexible strategy $w_1$	Change flex strategy $w_2$	External flex strategy $w_3$	TCOST
$p = 0.5$			
29.4%	0	70.6%	267.4
29.4%	36.1%	34.5%	245
$p = 0.7$			
51.1%	0	48.9%	240.4
51.1%	34.8%	14.1%	235.7

TABLE 4: Solving units for different diversities. Unit: ten thousand yuan.

$a = 30, b = 150, c = 5, d = 10, e = 80, f = 56, g = 30$			
$S = 1, p = 0.5, r = 0.1$			
Use a flexible strategy $w_1$	Change flex strategy $w_2$	External flex strategy $w_3$	TCOST
$v = 0.6$			
29.4%	0	70.6%	267.4
29.4%	36.1%	34.5%	245
$v = 0.4$			
21.1%	0	78.9%	281.6
21.1%	31.6%	47.3%	268.9

strategies increased from 21.1% to 29.4%. It can be seen that the more uneven the distribution of diversity, the more flexible the cloud provider needs to use to reduce costs

- (2) In the case when the flexible strategy was changed, a minimum value of 245 would occur. However, when a flexible strategy was changed, a minimum value of 268.9 would follow. When diversity was increased from 0.4 to 0.6, the proportion of investment using flexible strategies increased from 21.1% to 29.4%, indicating a growth of 8.3%, and the proportion of investment changing flexible strategies increased from 32.6% to 36.1%, implying a growth of 3.5%. It can be seen that the more uneven the distribution of diversity, the higher the proportion of investment increases than the proportion of investment, and

the more cloud providers can consider the investment using the flexible strategy

## 6. Summary and Prospect

Our major findings provide multiple insights: (1) the more cloud providers adopt the changed flexible strategy, the more conducive it is to reduce costs [7]; (2) the larger the application load, the more cloud providers need to use flexible strategies to lower costs [8]; (3) the less the impact of changing the flexible strategy on costs, the more conducive cloud providers use the flexible strategy to decrease costs; (4) the more uneven the distribution of diversity, the higher proportion of investment increases than the proportion of investment, and the more cloud providers consider the investment using flexible strategy. According to the commercial characteristics

of cloud-native architecture with cloud service as the delivery model, this paper incorporates the percentage of normal service time into the external impacting factors of the flexible cost investment strategy of public health organizations and fully considers the compensation cost caused when SLA cannot be satisfied in the proposed computational model. Data in practice is used to discuss the different cost investment strategies of cloud providers when each external influencing factor changes. The results provide a reference for public health information system cloud providers to develop reasonable flexible cost investment strategy optimization sets under different environments and needs and expand the research category in information system cost.

In this paper, the commercial characteristics of cloud-native architecture are fully considered, but the conversion value of normal service time percentage only hinged on Amazon and Microsoft's SLA. Hence, the lack of versatility is a deficiency of this study. And how to establish the corresponding relationship between normal service time percentage and conversion value in the system is another problem that needs further study in the future.

### Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

We declare that there is no conflict of interest regarding the publication of this paper.

### Acknowledgments

This article was supported by the Fujian Provincial Social Science Planning Project in 2020: Research on Capacity Building of Regional Medical Collaborative Innovation Community under the Background of "Internet + Medical" (No. FJ2020B038).

### References

- [1] C. Weinhardt, A. Anandasivam, B. Blau et al., "Cloud computing – a classification, business models, and research directions," *Business & Information Systems Engineering*, vol. 1, no. 5, pp. 391–399, 2009.
- [2] X. Liu, D. He, L. Yang, and C. Liu, "A novel negative feedback information dissemination model based on online social network," *Physica A: Statistical Mechanics and its Applications*, vol. 513, pp. 371–389, 2019.
- [3] C. Sien and L. Zhenyu, "Research on flexible cost generation mechanism of software architecture based on business process perception," *Modern Management Science*, vol. 12, no. 5, pp. 93–96, 2013.
- [4] V. Chang, "The business intelligence as a service in the cloud," *Future Generation Computer Systems*, vol. 37, no. C, pp. 512–534, 2014.
- [5] B. Mutschler, J. Bumiller, and M. Reichert, "Why process-orientation is scarce: an empirical study of process-oriented information systems in the automotive industry," in *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, pp. 433–440, Hong Kong, China, October 2006.
- [6] B. Clark, S. Devnanichulani, and B. Boehm, "Calibrating the COCOMO II post-architecture model," in *Proceedings of the 20th International Conference on Software Engineering*, pp. 477–480, Kyoto, Japan, 1998.
- [7] Z. Guo, J. Li, and R. Ramesh, "Optimal management of virtual infrastructures under flexible cloud service agreements," *Information Systems Research*, vol. 30, no. 4, pp. 1424–1446, 2019.
- [8] O. Rejeb, C. Pilet, S. Hamana et al., "Performance and cost evaluation of health information systems using micro-costing and discrete-event simulation," *Health Care Management Science*, vol. 21, no. 2, pp. 204–223, 2018.
- [9] W. N. Robinson and S. D. Pawlowski, "Managing requirements inconsistency with development goal monitors," *IEEE Transactions on Software Engineering*, vol. 25, no. 6, pp. 816–835, 1999.
- [10] S. Mahesh, B. J. L. Landry, T. Sridhar, and K. R. Walsh, "A decision table for the cloud computing decision in small business," *Information Resources Management Journal*, vol. 24, no. 3, pp. 9–25, 2011.
- [11] J. Gebauer, M. J. Shaw, and M. L. Gribbins, "Task-technology fit for mobile information systems," *Journal of Information Technology*, vol. 25, no. 3, pp. 259–272, 2010.
- [12] J. Gebauer and F. Schober, "Information system flexibility and the cost efficiency of business processes," *Journal of the Association for Information Systems*, vol. 7, no. 3, pp. 122–147, 2006.
- [13] M. Gelderman, "Task difficulty, task variability and satisfaction with management support systems," *Information & Management*, vol. 39, no. 7, pp. 593–604, 2002.
- [14] D. Q. Chen, M. Mocker, D. S. Preston, and Teubner, "Information systems strategy: reconceptualization, measurement, and implications," *MIS Quarterly*, vol. 34, no. 2, pp. 233–259, 2010.
- [15] J. Braa, O. Hanseth, A. Heywood, Mohammed, and Shaw, "Developing health information systems in developing countries: the flexible standards strategy," *MIS Quarterly*, vol. 31, no. 2, pp. 381–402, 2007.
- [16] J. Gebauer and P. Karhade, "A framework to assess the impact of information systems on the flexibility and performance of business processes," in *ICIS 2002 PROCEEDINGS*, pp. 320–440, USA, 2004.
- [17] J. S. Evans, "Strategic flexibility for high technology manoeuvres: a conceptual framework," *Journal of Management Studies*, vol. 28, no. 1, pp. 69–89, 1991.