Vol.2021-IOT-52 No.11
2021/3/1

家庭向けの遠隔ヘルスケアにおける DNS を活用した監視システムの試作

陸 子健 [1,a)]　金 勇 [2,b)]　山井 成良 [1,c)]　友石 正彦 [2,d)]

概要：IoT（Internet of Things）社会の発展に伴い，インターネットに接続される IoT 機器の増加が進んでいる．特に，遠隔医療分野では医療関係者や患者だけでなく，ヘルスケアにおける一般ユーザの数も年々増えつつある．しかし，これまでの遠隔医療やヘルスケアサービスの実現方法には，個人情報や医療関連データなどのプライバシー暴露の問題が懸念されている．本研究では，家庭向けのヘルスケアにおいてスケーラビリティとセキュリティ両方を考慮し，DNS を活用した遠隔監視システムを提案・試作する．

キーワード：IOT，DNS，ヘルスケア，遠隔監視

Prototype of Monitoring System for Remote Home Healthcare Using DNS

1. はじめに

IoT（Internet of Things） 社会の発展に伴い，インターネットに接続される IoT 機器の増加が進んでいる．また，遠隔医療分野では医療関係者や患者だけでなく，ヘルスケアにおける一般ユーザの数も年々増えつつある．特に新型コロナウイルスが世界的に流行っている現在には，自宅から離れずに遠隔医療サービスに対する需要が高まりつつある．しかし，これまでの遠隔医療やヘルスケアサービスの実現方法には，個人情報や医療関連データなどのプライバシー暴露の問題が懸念されている．

従来の遠隔ヘルスケアにおける監視手法では，一般的に IoT 機器を直接インターネットに接続し，ユーザからのデータの照会などの管理をインターネット経由で行われるのが一般的である．また，IoT 機器の莫大な増加に伴い，パブリッククラウド環境を活用することでスケーラビリティの実現を図っている．さらに，個人情報や医療関連データを保護するために，遠隔監視におけるユーザ認証とデータの暗号化が一般的なセキュリティ対策として考えられている．しかし，莫大な IoT 機器を扱うためのスケーラビリティ対策と個人情報や医療関連データを保護するためのセキュリティ対策両方を考慮するのが困難である．特に，パブリッククラウドを利用することにより，クラウドサービスプロバイダー側での個人情報漏洩やプライバシー暴露が懸念されている．そのため，IoT 環境において，特に遠隔医療やヘルスケアを対象としたシステムでは，スケーラビリティとセキュリティ両方を考慮することが求められている．

本研究では遠隔ヘルスケアにおいて，スケーラビリティとセキュリティ両方を考慮した軽量かつ安全な IoT システムの構築を目的とし，DNS（Domain Name System）を活用した遠隔監視手法を提案する．

2. 関連研究

従来の IoT システムにおける遠隔監視手法では，照会用端末がインターネットを経由して IoT 機器に直接アクセスし関連データを取得する方法が主に利用されている．そのため，照会用端末と IoT 機器間通信の安全性だけでなく，ユーザのプライバシーを保護するための IoT データの暗号化が求められている．すなわち，膨大な IoT 機器を扱うためのスケーラビリティと通信の安全性及びプライバシー保護を全て考慮する必要がある．

Shaheena らの研究では，遠隔ヘルスケアシステムにおいて，ユーザと医療従事者間の通信の安全性とプライバシー保護を考慮し，相互認証に基づいた鍵共有プロトコルを提案している[1]．この提案手法では相互認証に成功し鍵共有ができたユーザと医療従事者間のみ通信可能し，データサーバへのアクセスが可能になる．Anass らの研究では，共通鍵暗号化に基づいた IoT 機器とヘルスケアセンター間の安全な通信手法を提案している[2]．この提案手法では，セッション鍵管理システムと特殊なユーザ ID による認証方法を利用しており，全ての IoT 機器がユニークな ID を持ち秘密にデータベースに保存される．なお，IoT 機器の ID は平文で転送されないようにしている．Junaid らの研究ではアンドロイドプラットフォームをベースにしたモバイルアプリケーションを構築し，IoT による遠隔ヘルスケアシステムを提案している[3]．この提案システムは web サービ

[1] 東京農工大学
Tokyo University of Agriculture and Technology, Koganei, Tokyo, Japan
[2] 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo, Japan
[a)] KanoRikuo@net.cs.tuat.ac.jp
[b)] yongj@gsic.titech.ac.jp
[c)] nyamai@cc.tuat.ac.jp
[d)] tomoishi@noc.titech.ac.jp

スとクラウドシステムを利用してユーザデータを管理している. Jie らはブロックチェーン技術を用いた遠隔ヘルスケア監視システムを提案している[4]．この提案では，ブロックチェインに IoT データのハッシュ値を保存し，実データは暗号化された後に他の分散ファイルシステムに保存される．また，きめ細かいアクセスコントロールポリシーを導入し，許可されたユーザのみ IoT データへのアクセスが可能になるように管理している. Xiaohui らの研究ではユーザと医療従事者間の通信に着目し，プライバシー保護を考慮した遠隔ヘルスケア監視システムを提案している[5]．この提案では，ユーザの属性を使った認証方法を利用してユーザと医療従事者間の通信でユーザ情報が漏れないように工夫している．

これらの先行研究では，照会用端末がインターネットを介して IoT 機器あるいはホームネットワークに直接アクセスすることで IoT データを取得するため，膨大な数の IoT 機器に対応するためのスケーラビリティの課題がある．また，これらの先行研究では照会用端末がインターネットを介して IoT 機器にアクセスする際の通信路での暗号化や安全なユーザ認証が主な目的となっており，IoT データの暗号化は考慮しているものの，クラウド環境利用におけるプライバシー保護への考慮は不十分である．本研究では，家庭向けの遠隔ヘルスケアシステムにおいて，DNS を活用してスケーラビリティとセキュリティ両方を考慮し，照会用端末が IoT 機器あるいはホームネットワークに直接アクセスすることなく，クラウド環境の利用においてもプライバシー保護が可能な監視手法を提案する．

3. 提案手法

3.1 DNS を用いた監視手法

インターネットにおいて最も大規模な分散データベースの 1 つとして知られている DNS には，目的によって SOA，A，NS，TXT などなさまざまなレコードが設定できる．本研究ではテキストデータとバイナリデータ 2 種類のデータを対象とする．テキストデータは，体重計のような IoT 機器から IoT 通信プロトコルにより IoT ゲートウェイが取得し DNS サーバへアップロードする．また，写真のようなバイナリデータは, 小型のデータベースに保存される．

IoT 機器へ直接アクセスしデータを得る従来の方式と違い，本提案手法ではテキストデータはもちろん，バイナリデータも DNS を用いてコンテナ技術とデータベース技術と連携しデータを取得することとする. DNS の大規模な分散データベースの特徴を利用し，ホームネットワークとパブリッククラウドを，プライマリ DNS サーバ 1 台とセカンド DNS サーバ 1 台以上，2 台以上 DNS サーバを設置しデータを保存する．

3.2 データの照会と共有

2 台 DNS サーバを設置することによりデータ不一致問題を避けるため DNS ゾーン転送を利用する．ホームネットワークにある DNS サーバはプライマリサーバ，パブリッククラウドにある DNS サーバはセカンドリサーバとする．既にプライマリ DNS サーバにアップロードしているデータを，DNS ゾーン転送によってセカンドリ DNS サーバと同期させる. また, プライマリ DNS サーバとセカンドリ DNS サーバ間の同期はプライマリ DNS サーバに更新があった場合或いは一定期間の間隔を開けて行うことになる.

本提案システムでの IoT 機器のデータ照会は，ホームネットワーク内ではプライマリ DNS サーバに問い合わせる方法で実現可能である．また，ホームネットワーク内のプライマリ DNS サーバとパブリッククラウド上のセカンドリ DNS サーバ間の同期により，認証機能を導入し許可されたユーザのみセカンドリ DNS サーバに問い合わせすることによって IoT 機器のデータ照会が可能になる．

本提案システムにより照会・共有可能なデータは IoT 機器から取得するテキストデータとカメラやアプリケーションから取得するバイナリデータ（写真，ファイルなど）に分けられる．IoT 機器から取得可能なテキストデータは DNS システムに TXT レコードとして登録することで照会可能である．一方バイナリデータは DNS システム上で扱えないため他のデータベースシステムの利用を考える．ただし，軽量かつ安全性を考慮し，DNS システムとコンテナ技術（Docker など）を連携することにより，許可されたユーザからバイナリデータへの照会リクエストに応じてセカンドリ DNS サーバ経由でコンテナを作成し，ユーザが照会可能にする．さらに，ユーザの照会終了後には一定時間後にそのコンテナを削除することで不正アクセスなどを防ぐことが可能である．

3.3 スケーラビリティ

日々増加しつつある膨大の数の IoT 機器をいかにして同時に扱うことが本研究におけるスケーラビリティ対策の主な課題である．本研究では，インターネットにおいて DNS サーバは数が多く，設置しやすい，負荷軽い，さまざまなメリットがあり，また，本研究の対象においてデーは主にテキストデータと考えている上，既に実績があり広く使われている技術としてスケーラビリティ対策に期待できる．また，DNS システムはインターネット上のドメイン名の管理を行うプラットフォームであり，例えば「.com」のドメイン名ではおよそ 1 億 5000 万個のサブドメイン名が同時に管理されている．このような DNS システムを活用することにより，軽量でスケーラブルな遠隔ヘルスケアシステムの実現が期待できる．

3.4 セキュリティとプライバシー保護

本研究で考慮されているセキュリティ対策には IoT 機器を遠隔で照会する際のユーザ認証機能と IoT 機器のデータの暗号化が含まれている．本提案システムはホームネットワークを単位として構成し，インターネット上にパブリッ

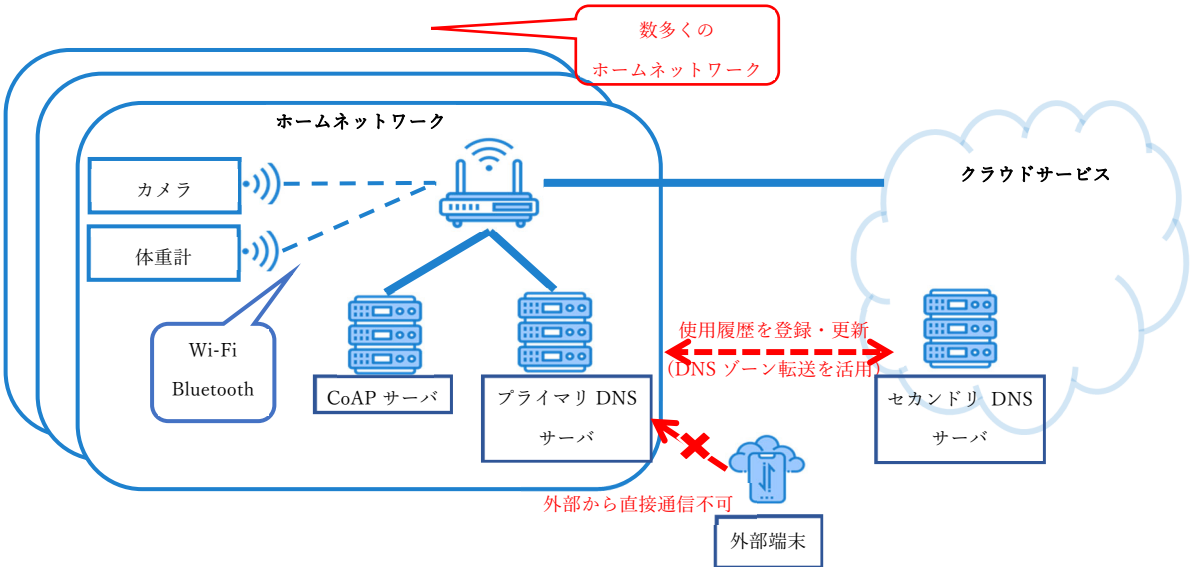

図 1　ホームネットワークの構成

ククラウドサービスを利用した照会用サーバを構築する．IoT 機器間の通信はそれぞれのホームネットワーク内で行われ，暗号化された IoT データのみインターネット上のクラウドシステムに設置した照会用サーバに置くことになる．さらに，インターネットからホームネットワークへの直接通信は不可とし，ホームネットワークからインターネットへの通信は DNS サーバ間のゾーン転送のみとする．その際に，DNS システムの認識機能（TSIG）を使用し，DNS サーバに保存される IoT データは暗号化されたデータのみとする．

ホームネットワークは世帯を単位として構成し，その中で各 IoT 機器からのデータはホームネットワーク内のプライマリ DNS サーバに DNS レコードで登録される．ホームネットワークにある端末から各 IoT 機器へのアクセスは可能であるが，インターネットからの外部端末は直接アクセスできないように制限される．しかし，インターネット上で許可されたユーザが各ホームネットワーク内の IoT 機器のデータを照会することが可能であるが，直接ホームネットワーク内にアクセスするのではなく，DNS サーバ間のゾーン転送により IoT 機器のデータ同期が行われたパブリッククラウドにあるセカンドリ DNS サーバに問い合わせることで実現可能である．

4. 実験と結果

4.1 ホームネットワークの構築

本提案システムはネットワークによって二つの部分が分けられている．図 1 のように，ホームネットワークは IoT 機器, IoT サーバ, プライマリ DNS サーバ, データベース, 他のネットワークデバイスが含まれている．

4.2 CoAP による IoT 機器間通信

本研究では RFC 7252 で定義されている Constrained Application Protocol （CoAP）[6]を使い，体重計や写真機能付きの IoT 機器などから，テキストデータとバイナリデータを生成し CoAP を使って IoT 機器が CoAP サーバと通信することになる．ユーザが IoT 機器を使用後，IoT 機器が CoAP サーバと通信し，CoAP サーバが IoT 機器から受信したデータをホームネットワークにあるプライマリ DNS サーバか小型データベースに保存する．

IoT 機器は様々なデータを生成するが，タイプはテキストとバイナリ二種類が分けられる．本研究においては，データ生成シミュレーター，クラウドとしてツール Copper [7]を使用する．それによって CoAP サーバに PUT メソッドで通信する．図 2 のように, Copper で登録されている IoT 機器体重計とカメラが検出されている．体重計から「id:52;date:20210104;device:weight;user:root;data:59.9;」のようなテキストデータを PUT メソッドで CoAP サーバに送信する．そして，図 3 に CoAP サーバから ACK タイプの確認済み返事を返すことを示す．

図 4 と図 5 には, バイナリデータはカメラから「camera1.jpg」を PUT メソッドで CoAP サーバに送信する例を示す. CoAP が UDP を使用するので, ここでは CoAP パケットを 512Byte と設定する．バイナリデータは 512Btye を超えることがよくあるので複数のパケットを分割し送る．なお，PUT リクエストごとに ACK 確認が得られる．そして，データがデーターベースに保存される．

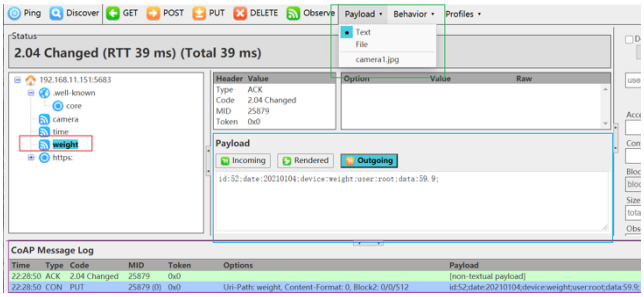

図 2　テキストデータを送信する例

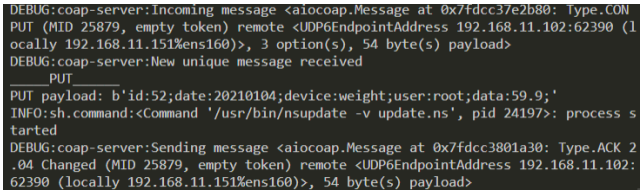

図 3　CoAP サーバによる ACK 応答

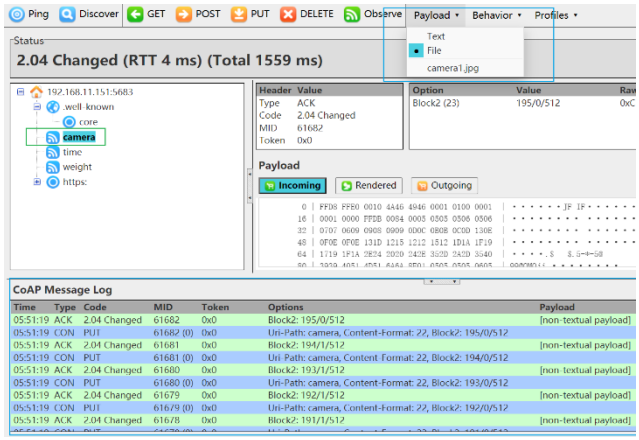

図 4　バイナリデータを送信する例

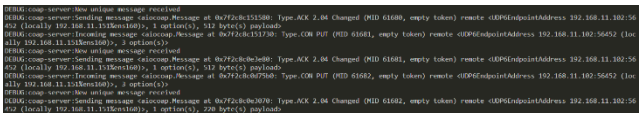

図 5　たくさんの分割したパケット

4.3 DNS への IoT データ登録

IoT 機器からのデータは CoAP サーバ経由して，テキストデータはプライマリ DNS サーバにテキストレコード（TXT タイプ）として登録される．バイナリデータは，小型データーベースにバイナリタイプそのまま保存されると同時に，データーベースからデータを取り出すため SRV タイプとして登録される．ここでは，データが特定できる情報を，「日付 date」「IoT 機器 device」「ユーザ user」として，「date.device.user.myhome.healthcare.net. 」という形で使用する．

4.4 DNS ゾーン転送

IoT 機器と CoAP サーバから最新 DNS レコードが登録されると，ゾーン転送によるプライマリ DNS サーバとセカンドリ DNS サーバ間の同期が行われる．

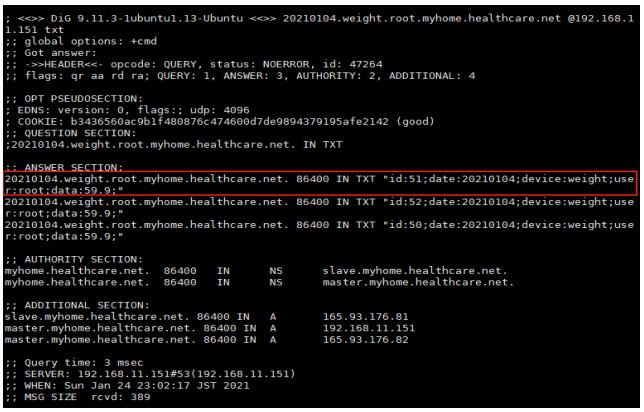

図 6　TXT タイプレコードを検証

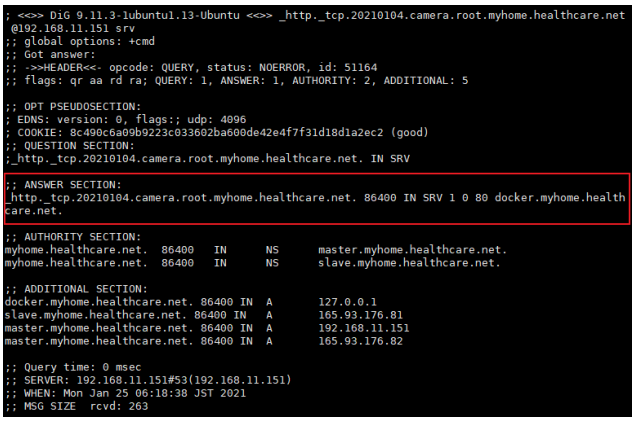

図 7　SRV タイプレコードを検証

ただし，外部からホームネットワークへの通信が禁止されているが，ホームネットワークからの DNS ゾーン転送通信だけはできるようになっている．テキストデータか或いはバイナリデータは CoAP サーバに届いた後，CoAP サーバがクライアントに ACK 送る前に TXT タイプか SRV タイプとしてプライマリ DNS サーバに送って登録されるが，ホームネットワーク外にあるセカンド DNS サーバに Notify メリットを送って差分情報を，つまり最新データだけ登録する．

4.5 クラウド側ネットワークの構築

図 8 にクライアント側ネットワークの構成を示す．中には照会用のセカンドリ DNS サーバ，セカンドリ DNS サーバに問い合わせする前にパケットを処理するプロキシサーバ，バイナリデータ取得用，HTTP サービス提供する一時的に起動されるコンテナなど含まれている．ただし，本研究ではプロキシサーバとセカンドリ DNS サーバ同じマシンで稼働される．なお，外部からホームネットワークへのアクセスは全て禁止とする．

4.5.1 コンテナ技術

テキストデータのように DNS サーバに保存できないバイナリデータに対して，コンテナ技術とデータベースと連携するという対策を行う．HTTP サーバが含まれるコンテナは，常時に動いている一般的な HTTP サーバとは違い，必要な機能だけ含むので消耗が抑えられている一方，生存時間を設定することによって起動後生存時間を超えると自動的に削除することで不正アクセスなどを防ぐことができる．バイナリデータが表示できるように，本研究では取ったデータをブラウザで表示とする．

4.5.2 クライアントからのデータ照会・共有

照会リクエストが来ると，DNS プロキシによってデータ照会用 HTTP サーバに内蔵されているコンテナ新しく作る．また，新たなコンテナが 120 秒後自動的に削除される．なお，クライアントからリクエストに応じて，一つのリクエストごとにコンテナが新しく作られる．

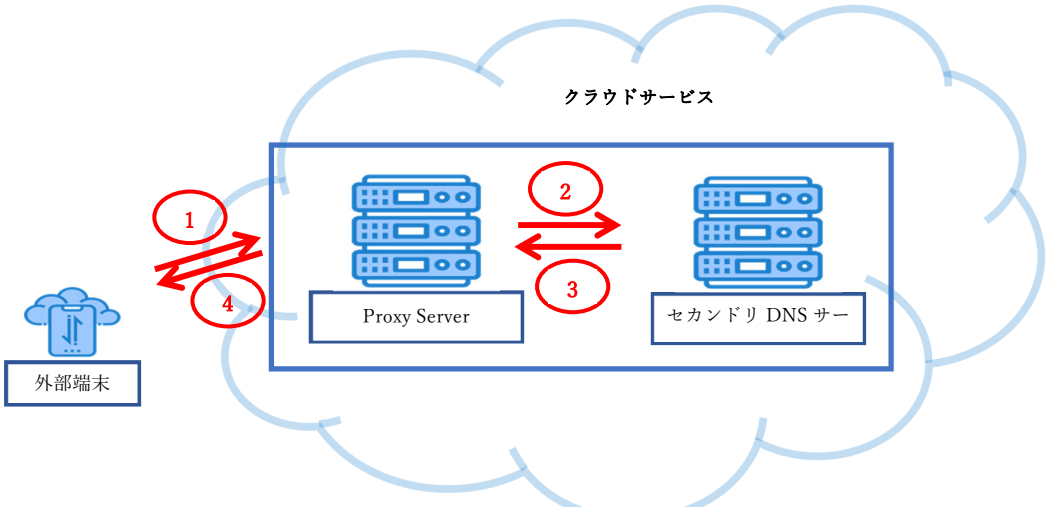

図 8　ホームネットワーク外の構成及びテキストデータ照会の流れ

4.5.3 テキストデータ照会

図 8 にはテキストデータ照会に流れを示す. 図 9 は SETP 3 のセカンド DNS サーバからの応答．図 10 はプロキシサーバによる転送.

- STEP 1 外部端末から,「日付 20210104」「user root」「IoT 機器 体重計」というクエリを投げる．ここでは dig でクエリをシミュレーションとする.
- STEP 2 投げられて来るクエリは，プロキシサーバがそのパケットをアンパックして，TXT か SRV かクエリのタイプを確認してから再びパックして，セカンドリ DNS サーバに転送.
- STEP 3 セカンドリ DNS サーバが TXT タイプクエリに応じてプロキシサーバーに応答する.
- STEP 4 プロキシサーバはそのままクライアントに転送

4.5.4 バイナリデータ照会

図 11 はバイナリデータ照会する流れ．図 11 はプロキシサーバによる SRV タイプクエリを転送する．図 13 はコンテナ起動の検証．図 14 には結果検証を示す。

- STEP 1 外部端末から,「日付 20210104」「user root」「IoT 機器 カメラ」というクエリを送信する. ここでは dig でクエリをシミュレーションとする.
- STEP 2 投げられて来るクエリは，プロキシサーバがそのパケットをアンパックして，TXT か SRV かクエリのタイプを確認し，小型 HTTP サーバが内蔵される生存時間 120 秒のコンテナを作てから再びパックして，セカンドリ DNS サーバに転送.
- STEP 3 セカンドリ DNS サーバが SRV タイプクエリに応じて，HTTP サーバの IP アドレスとポート番号が含まれる SRV 応答をプロキシサーバーに返す.
- STEP 4 プロキシサーバはそのままクライアントに転送.

```
; <<>> DiG 9.11.3-1ubuntu1.13-Ubuntu <<>> 20210104.weight.root.myhome.healthcare.net @165.93.176.81 txt
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39223
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8cc812291c8d3a8017d8b81c600dc80b0016fd3f83cb1fc8 (good)
;; QUESTION SECTION:
;20210104.weight.root.myhome.healthcare.net. IN TXT

;; ANSWER SECTION:
20210104.weight.root.myhome.healthcare.net. 86400 IN TXT "id:50;date:20210104;device:weight;user:root;data:59.9;"
20210104.weight.root.myhome.healthcare.net. 86400 IN TXT "id:51;date:20210104;device:weight;user:root;data:59.9;"
20210104.weight.root.myhome.healthcare.net. 86400 IN TXT "id:52;date:20210104;device:weight;user:root;data:59.9;"

;; AUTHORITY SECTION:
myhome.healthcare.net.  86400   IN      NS      slave.myhome.healthcare.net.
myhome.healthcare.net.  86400   IN      NS      master.myhome.healthcare.net.

;; ADDITIONAL SECTION:
slave.myhome.healthcare.net. 86400 IN   A       165.93.176.81
master.myhome.healthcare.net. 86400 IN  A       165.93.176.82
master.myhome.healthcare.net. 86400 IN  A       192.168.11.151

;; Query time: 2 msec
;; SERVER: 165.93.176.81#53(165.93.176.81)
;; WHEN: Mon Jan 25 04:18:34 JST 2021
;; MSG SIZE  rcvd: 389
```

図 9　セカンドリ DNS サーバに送信と応答

```
Request: [165.93.176.82:46201] (udp) / '20210104.weight.root.myhome.healthcare.net.' (TXT)
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39223
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 2, ADDITIONAL: 4
;; QUESTION SECTION:
;20210104.weight.root.myhome.healthcare.net. IN      TXT
;; ANSWER SECTION:
20210104.weight.root.myhome.healthcare.net. 86400   IN      TXT     "id:50;date:20210104;device:weight;user:root;data:59.9;"
20210104.weight.root.myhome.healthcare.net. 86400   IN      TXT     "id:51;date:20210104;device:weight;user:root;data:59.9;"
20210104.weight.root.myhome.healthcare.net. 86400   IN      TXT     "id:52;date:20210104;device:weight;user:root;data:59.9;"
;; AUTHORITY SECTION:
myhome.healthcare.net.  86400   IN      NS      slave.myhome.healthcare.net.
myhome.healthcare.net.  86400   IN      NS      master.myhome.healthcare.net.
;; ADDITIONAL SECTION:
slave.myhome.healthcare.net. 86400   IN      A       165.93.176.81
master.myhome.healthcare.net. 86400   IN      A       165.93.176.82
master.myhome.healthcare.net. 86400   IN      A       192.168.11.151
;; OPT PSEUDOSECTION
; EDNS: version: 0, flags: ; udp: 4096
; EDNS: code: 10; data: 8cc812291c8d3a8017d8b81c600dc80b0016fd3f83cb1fc8
Reply: [165.93.176.82:46201] (udp) / '20210104.weight.root.myhome.healthcare.net.' (TXT) / RRs: TXT,TXT,TXT
```

図 10　TXT タイプ転送

- STEP 5 クライアントはもらった SRV 応答にある IP アドレスとポート番号によってコンテナにある HTTP サーバにアクセスしバイナリデータを取得．ここでは，データ検証はブラウザで行う.

5. 今後の課題

5.1 データの暗号化と認証機能

本提案システムはデータの暗号化と認証機能を導入せず行ったが，テキストデータとバイナリデータそのまま保存する．バイナリデータはバイナリタイプであるが，分析できるか解読できるツールがあれば見られる恐れがあり，また，DNS サーバに保存されるテキストデータはそのまま人間にも読める内容なので見られると重大な問題が生じると考えられる.

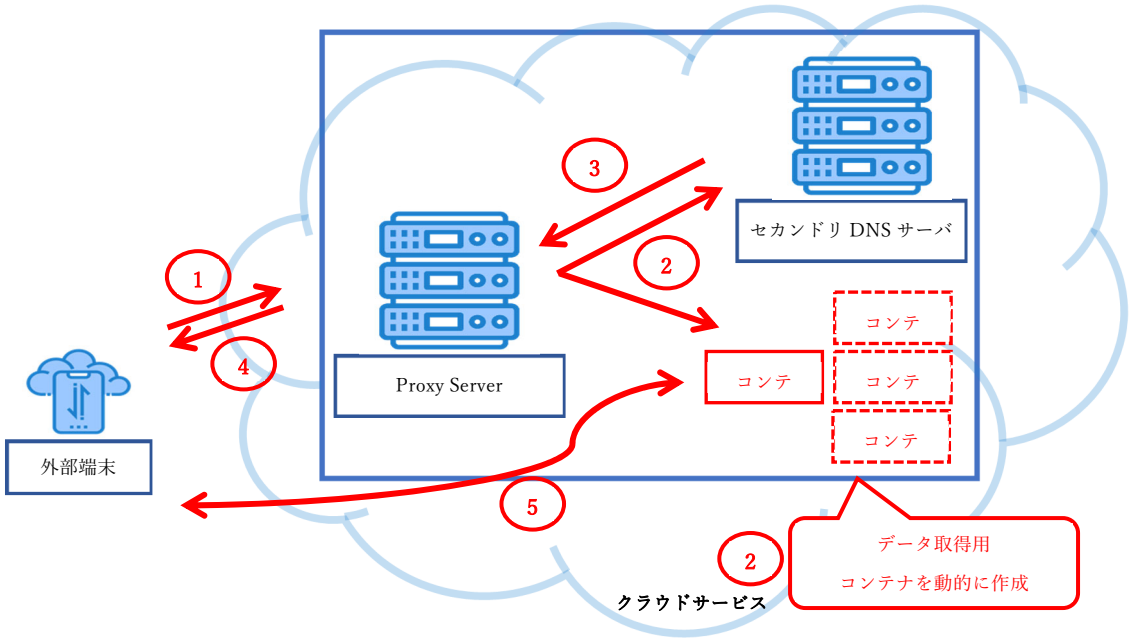

図 11 ホームネットワーク外の構成及びバイナリデータ照会用の流れ

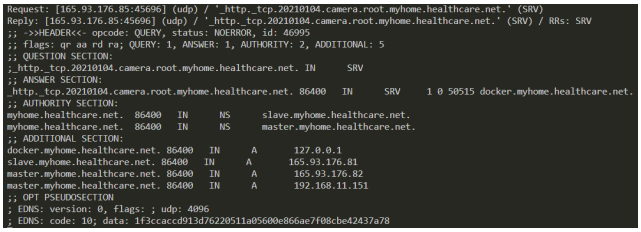

図 12 SRV タイプ転送

図 13 コンテナの起動

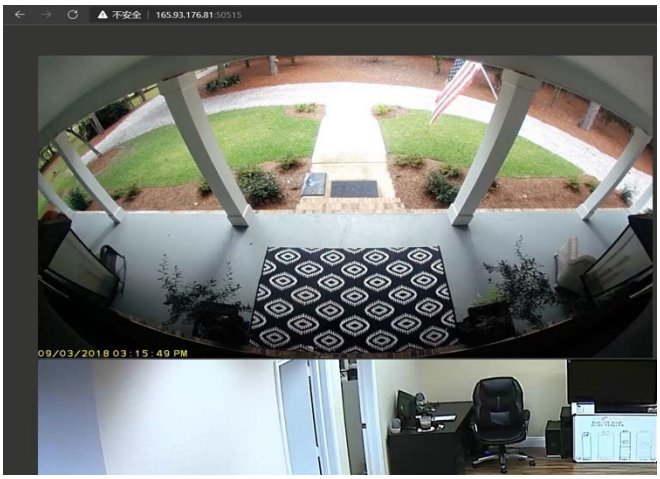

図 14 データの取得と表示

参考文献

[1] S. Khatoon, S. M. M. Rahman, M. Alrubaian and A. Alamri, "Privacy-Preserved, Provable Secure, Mutually Authenticated Key Agreement Protocol for Healthcare in a Smart City Environment," in IEEE Access, vol. 7, pp. 47962-47971, 2019, doi: 10.1109/ACCESS.2019.2909556.

[2] A. Rghioui, A. L'aarje, F. Elouaai and M. Bouhorma, "The Internet of Things for healthcare monitoring: Security review and proposed solution," 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, 2014, pp. 384-389, doi: 10.1109/CIST.2014.7016651.

[3] J. Mohammed, C. Lung, A. Ocneanu, A. Thakral, C. Jones and A. Adler, "Internet of Things: Remote Patient Monitoring Using Web Services and Cloud Computing," 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), Taipei, 2014, pp. 256-263, doi: 10.1109/iThings.2014.45

[4] J. Xu et al., "Healthchain: A Blockchain-Based Privacy Preserving Scheme for Large-Scale Health Data," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8770-8781, Oct. 2019, doi: 10.1109/JIOT.2019.2923525.

[5] X. Liang, X. Li, R. Lu, X. Lin and X. Shen, "Enabling pervasive healthcare with privacy preservation in smart community," 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, 2012, pp. 3451-3455, doi: 10.1109/ICC.2012.6363888.

[6] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014.

[7] “Copper for Chrome (Cu4Cr) CoAP user-agent” (online), available from https://github.com/mkovatsc/Copper4Cr (accessed 2021-01-05)